

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Саратовский государственный технический университет имени Гагарина Ю.А.»

Филиал федерального государственного бюджетного образовательного
учреждения высшего образования
«Саратовский государственный технический университет имени Гагарина Ю.А.»
в г. Петровске

 УТВЕРЖДАЮ
Директор филиала СГТУ
имени Гагарина Ю.А. в г.Петровске
Е.А.Бесшапошникова
«30» июня 2021 г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

по дисциплине

ОП.11 «Вычислительная техника»

специальности

13.02.07 «Электроснабжение (по отраслям)»

Методические указания рассмотрены
на заседании предметной (цикловой)
комиссии общепрофессиональных
дисциплин, профессиональных модулей
специальностей
технического профиля

«14» июня 2021 года, протокол № 13

Председатель ПЦК  /Т.А.Лескина/

Петровск 2021

Пояснительная записка

Методические указания по выполнению практических работ разработаны в соответствии с рабочей программой учебной дисциплины «Вычислительная техника», требованиями Федерального государственного образовательного стандарта (далее – ФГОС) по специальности среднего профессионального образования (далее – СПО) 13.02.07 Электроснабжение (по отраслям), утвержденного приказом Министерства образования и науки РФ от 14.12.2017 № 1216 и соответствующих общих (ОК) и профессиональных (ПК) компетенций:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 09. Использовать информационные технологии в профессиональной деятельности;

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 1.1. Выполнять основные виды работ по проектированию электроснабжения электротехнического и электротехнологического оборудования;

ПК 1.2. Читать и составлять электрические схемы электроснабжения электротехнического и электротехнологического оборудования.

ПК 2.5. Разрабатывать и оформлять технологическую и отчетную документацию.

Цели изучения дисциплины «Вычислительная техника»:

- формирование профессиональных компетенций;
- использование полученных знаний, умений и навыков применения вычислительной техники и сетей в процессе изучения профессиональных дисциплин и в дальнейшей профессиональной деятельности;
- ознакомление с основами современных компьютерных коммуникаций и тенденциями их развития.

При выполнении практических работ студент должен **знать**:

- способы представления информации в ЭВМ;
- основные логические элементы цифровой техники;
- составные части, общие принципы организации и функционирования компьютерных систем;

- архитектуру процессоров;

- принцип работы основной памяти и периферийных устройств.

Содержание практических занятий определено рабочей программой и тематическим планированием, соответствует теоретическому материалу изучаемых разделов учебной дисциплины.

Объём практических занятий по дисциплине определяется учебным планом по данной специальности.

Продолжительность практического занятия - 2 академических часа. Перед проведением практического занятия преподавателем организуется инструктаж, а по ее окончании – обсуждение итогов.

Комплект методических указаний по выполнению практических работ дисциплины «Вычислительная техника» содержит 11 практических занятий.

**Перечень практических работ
по дисциплине «Вычислительная техника»**

ПРАКТИЧЕСКАЯ РАБОТА №1

Тема: Формы представления логических функций

ПРАКТИЧЕСКАЯ РАБОТА №2

Тема: Закон алгебры логики.

ПРАКТИЧЕСКАЯ РАБОТА №3

Тема: Основные логические элементы.

ПРАКТИЧЕСКАЯ РАБОТА №4

Тема: Триггеры.

ПРАКТИЧЕСКАЯ РАБОТА №5

Тема: Понятие архитектуры микропроцессора

ПРАКТИЧЕСКАЯ РАБОТА №6

Тема: Понятие архитектуры микропроцессора

ПРАКТИЧЕСКАЯ РАБОТА №7

Тема: Комбинационные схемы.

ПРАКТИЧЕСКАЯ РАБОТА №8

Тема: Конечные автоматы.

ПРАКТИЧЕСКАЯ РАБОТА №9

Тема: Работа арифметико-логических устройств

ПРАКТИЧЕСКАЯ РАБОТА №10

Тема: Алгоритмы управления

ПРАКТИЧЕСКАЯ РАБОТА №11

Тема: Процедура выполнения команд.

ИНСТРУКЦИИ ДЛЯ ОБУЧАЮЩИХСЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

Прежде чем приступить к выполнению заданий, внимательно прочитайте данные рекомендации. Практические работы включают в себя задания следующих видов:

Выполнение тестовых заданий

Для проверки и последующего анализа своих знаний Вам предлагается пройти тестовые задания. Выбор заданий осуществляется тестирующей системой случайным образом.

Тестовые задания интерактивны. По структуре формирования ответа различают следующие типы заданий:

- тесты восстановления соответствия - предусматривают восстановление соответствия между одинаковыми по величине, но различными по записи числами.
- тесты восстановления порядка - предусматривают расстановку чисел в соответствие с указанным порядком.
- тесты единственного выбора - предусматривают выбор одного правильного ответа из нескольких предложенных вариантов,
- тесты открытого типа - предусматривают ввод текстовых данных.

При вводе ответа необходимо соблюдать следующие правила:

- курсор нужно поместить в окно для ввода,
- вписывать слова нужно без сокращения,
- вписывать числовые выражения нужно без пробелов, строго следуя образцу, приведенному в задании.

Несоблюдение правил выполнения тестов открытого типа приведет к обозначению ответа как неверного.

Перед выполнением задания внимательно прочитайте его формулировку и предлагаемые варианты ответа. Отвечайте только после того, как Вы поняли вопрос и проанализировали все варианты ответа.

Выполняйте задания в том порядке, в котором они представлены в тесте. Выбор правильных ответов осуществляется путем выбора правильных ответов из списка.

Тестовые задания оцениваются в баллах. Все вопросы имеют свое балльное значение, что определяется, в первую очередь, сложностью самого вопроса.

Постарайтесь выполнить как можно больше заданий и набрать наибольшее количество баллов. По завершении тестирования баллы суммируются.

После выполнения тестовых заданий обязательно сохраните Ваши ответы и предоставьте их учителю.

Создание презентаций

ПРАВИЛА ПОСТРОЕНИЯ СОДЕРЖАНИЯ.

Правило 1. Содержание должно быть структурировано.

Содержание презентации должно быть четко структурировано: каждый новый слайд должен логически вытекать из предыдущего и одновременно подготавливать появление следующего. Лучший способ проверить, правильно ли построена презентация, — быстро прочитать только заголовки. Если после этого станет ясно, о чем презентация — значит, структура построена верно.

Правило 2. Краткость — сестра убедительности.

После того как содержание презентации собрано, с ним следует аккуратно поработать, сократив его насколько возможно. Оптимальным объемом презентации считается 24 традиционных слайда, если презентация уместится в 16 слайдов — еще лучше, ну а 12 и менее слайдов — это то, что редко встречается и крепко запоминается. В среднем, один слайд - это 1,5 минуты выступления.

ПРАВИЛА СОЗДАНИЯ СЛАЙДОВ.

Правило 1. Думать о зрителе.

При разработке формы презентации всегда следует думать о том, как зритель ее будет видеть. В первую очередь нужно решить, где зрители будут смотреть вашу презентацию: на бумаге, экране монитора или на большом экране с помощью проектора. На конкурс вы создаете презентации для экрана монитора! И возможно, вашу презентацию захотят распечатать. Это следует учитывать при выборе размера и цвета шрифтов.

Правило 2. Последовательность и единство оформления.

Все однотипные элементы должны всегда быть в одном месте: если зритель знает, где ждать заголовков, а где график, он лучше схватывает суть дела. Заголовок — всегда в одном месте экрана. График — всегда в одном месте экрана. И т.д. Однотипные подписи — одинакового цвета и размера. И т.д.

Правило 3. Нет тексту!

«Нет» любому тексту, кроме абсолютно необходимого. Читать страницу за страницей и запоминать текст совсем непросто. Количество текста на слайдах должно составить не более 35% от всего содержимого слайдов. Весь ненужный текст следует оставить либо для устного выступления (для текста доклада, т.к. у нас заочная конференция), либо заменить его графиками, картинками и т.д.

ВАЖНЫЕ ЗАПРЕТЫ.

1. Изображения и текст на слайдах не должны быть мелкими (даже если презентация готовится для экрана).

2. Если презентация будет цветной, то следует избегать ярких, так называемых чистых тонов — алого, ярко-синего, зеленого, фиолетового (они режут глаз). Такие краски следует зарезервировать для выделения действительно ключевых моментов, а для рядовых изображений использовать пастельные тона и контрастные сочетания цветов шрифта и фона.

3. Пестрота на экране (больше четырех цветов одновременно).

4. Самый главный запрет - спецэффекты. Анимации наподобие вращающихся заголовков, переворачивающихся слайдов, любые звуки - все это лишь отвлекает слушателей и необоснованно растягивает время презентации.

ОСНОВНЫЕ ПРАВИЛА ВЫСТУПЛЕНИЯ.

Презентация состоит из двух частей: демонстрация слайдов и сопровождение их текстом. Слайды — поддержка выступления, а не наоборот. Очень часто докладчик вместо выступления просто зачитывает текст на слайдах. Таких ораторов слушатели не уважают, текст они могут и сами прочитать.

Именно поэтому на конкурс мы обязательно требуем ТЕКСТ ДОКЛАДА.

Правило 1. Стройте выступление на аргументах, а не на слайдах.

Если презентация сделана правильно и текст хорошо сбалансирован другими визуальными элементами, то все равно не следует вести свою аудиторию по презентации, как экскурсовод туристов: «посмотрите налево, посмотрите направо». Презентер должен вести аудиторию не от слайда к слайду, а от тезиса к аргументу, от аргумента к примеру, от вывода к выводу. Нельзя говорить «перейдем на страницу 7», надо — «как именно мы решаем эту проблему, рассказывается на слайде 7». Нельзя говорить «посмотрите на следующий слайд», надо «и что же из этого следует? А вот что!» - и показываем слайд.

Правило 2. Готовьтесь к выступлению.

Выступление должно быть подготовлено, прорепетировано и отхронометрировано (подогнано под временные рамки).

Правило 3. Помните, что аудитория — это живые люди. Позволяйте себе эмоции.

Позволяйте себе в тексте восклицательные знаки. Текст вовсе не должен быть сухим! Вы не диктор ТВ, вы живой человек, который свято верит в то, о чем он рассказывает

Работа за компьютером

При любой работе должны соблюдаться определённые правила поведения и безопасности, чтобы сохранить своё здоровье и уберечься от возможных травм или каких-либо заболеваний. Профилактика лучше лечения, поэтому правила работы за компьютером необходимо знать всем, ведь мы всё больше и больше времени проводим именно за компьютером — за ним сидим на работе, и за ним же сидим дома.

Памятка ниже будет весьма полезна для людей всех возрастных категорий, чья жизнь или работа напрямую связана с ПК и на компьютере приходится долго и часто работать.

1. Сидите прямо.
2. Вам должно быть удобно. Но это не значит, что надо подгибать ноги под себя или класть ногу на ногу, сутулиться. Этого делать НЕЛЬЗЯ!
3. Верхняя часть монитора должна быть расположена на уровне глаз или чуть ниже, а нижняя чуть ближе к Вам.
4. Расстояние между монитором и глазами должно быть 45-75 см.
5. Освещение должно падать так же как и при писании с левой стороны, свет не должен быть сильно ярким или тусклым.
6. Не забывайте моргать, при моргании глаз омывается слёзной жидкостью и не пересыхает, а пересыхание глаза вредит зрению.

7. Периодически необходима зарядка для глаз, которую можно делать и на работе, и дома.
8. Каждый час работы за компьютером делайте перерыв на 15-20 минут.
9. Можете купить специальные очки для работы за ПК, их можно найти в каждой оптике.
10. Если Вы устали, началось чувство сонливости или тяжести в глазах, Вы не должны продолжать работу!
11. Обязательно каждый день надо проветривать комнату, вытирать пыль, влажная уборка только на пользу пойдёт.

ПРАКТИЧЕСКАЯ РАБОТА №1

Тема: Формы представления логических функций

Цель: научиться представлять логические функции

Оборудование, приборы, аппаратура, материалы: персональный компьютер с выходом в Интернет.

Справочный материал 1,2

Содержание работы

1. Организационный момент
 - Проверка готовности учащихся к уроку.
 - Приветствие.
 - Проверка готовности ребят к уроку
2. Постановка темы и цели урока
3. Повторение изученного материала

Основные понятия и определения.

Определение. Формула называется тождественно-истинной (тавтологией), если для любых наборов переменных она принимает значение И.

Определение. Формула называется тождественно тождественно-ложной, если для любых наборов переменных она принимает значение Л.

В алгебре высказываний используют две нормальные формы: дизъюнктивную и конъюнктивную нормальные формы формулы (ДНФ и КНФ).

Дизъюнктивной нормальной формой (ДНФ) называется дизъюнкция простых конъюнкций.

Конъюнктивной нормальной формой (КНФ) формулы есть формула, равносильная исходной формуле логики высказываний и записанная в виде конъюнкции элементарных дизъюнкций переменных.

Каждая формула, не равная тождественно Л, может быть приведена СДНФ, которая является единственной с точностью до перестановки дизъюнктивных членов.

Каждая формула, не равная тождественно И, может быть приведена к СКНФ, которая является единственной с точностью до перестановки конъюнктивных членов.

Совершенная дизъюнктивная нормальная форма формулы (СДНФ) это равносильная ей формула, представляющая собой дизъюнкцию элементарных конъюнкций, обладающая свойствами

1. Каждое логическое слагаемое формулы содержит все высказывания, входящие в формулу.
2. Все логические слагаемые формулы различны
3. Ни одно логическое слагаемое не содержит высказывание и его отрицание
4. Ни одно логическое слагаемое формулы не содержит одно и то же высказывание дважды.

Алгоритм получения СКНФ по таблице истинности:

- 1) Отметить те строки, в последнем столбце которых стоят 0:

- 2) Выписать для каждой отмеченной строки дизъюнкцию всех переменных следующим образом: если значение некоторой переменной в данной строке =0, то в дизъюнкцию включают саму эту переменную, если =1, то ее отрицание:
- 3) Все полученные дизъюнкции связать в конъюнкцию.

Образец решения

Построить таблицу истинности для высказывания: $(x \mid \bar{y}) \rightarrow (y \oplus z)$, построить СДНФ, СКНФ, найти минимальную ДНФ.

Решение.

Строим таблицу истинности- таблицу, с помощью которой устанавливается истинностное значение сложного высказывания при данных значениях входящих в него простых высказываний.

По таблице составляем дизъюнктивную нормальную форму (ДНФ). ДНФ в булевой логике — нормальная форма, в которой булева формула имеет вид дизъюнкции нескольких конъюнктов.

Алгоритм получения СДНФ по таблице истинности:

- 1) Отметить те строки, в последнем столбце которых стоят 1:
- 2) Выписать для каждой отмеченной строки конъюнкцию всех переменных следующим образом: если значение некоторой переменной в данной строке =1, то в конъюнкцию включают саму эту переменную, если =0, то ее отрицание:
- 3) Все полученные конъюнкции связать в дизъюнкцию:

Выбираем в таблице строки, в которых булева функция принимает значение 1. В данном случае – это 2-ая, 3-ая, 4-ая, 6-ая и 7-ая строки.

Для каждой строки составляем конъюнкцию: если значение переменной равно 0, то берем ее отрицание, а если 1, то берем саму переменную. Затем составляем дизъюнкцию полученных конъюнкций:

$$f(x, y, z) = (x \wedge \bar{y} \wedge z) \vee (x \wedge y \wedge \bar{z}) \vee (x \wedge \bar{y} \wedge \bar{z}) \vee (\bar{x} \wedge \bar{y} \wedge z) \vee (x \wedge \bar{y} \wedge z)$$

Выбираем в таблице строки, в которых булева функция принимает значение 0. В данном случае – это 1-ая, 5-ая, и 8-ая строки:

$$f(x, y, z) = (\bar{x} \wedge \bar{y} \wedge \bar{z}) \vee (x \wedge \bar{y} \wedge \bar{z}) \vee (x \wedge y \wedge z)$$

ДНФ называется минимальной, если она содержит наименьшее число букв среди всех ДНФ ей равносильных.

Метод Квайна основывается на применении двух основных соотношений.

Соотношение склеивания :

$$(a \wedge b) \vee (\bar{a} \wedge b) = b ; (a \vee b) \wedge (\bar{a} \vee b) = b$$

Соотношение поглощения:

$$a \wedge (a \vee b) = a \quad a \vee (a \wedge b) = a$$

Используя соотношение склеивания получаем:

$$(\bar{x} \wedge \bar{y} \wedge \bar{z}) \vee (x \wedge \bar{y} \wedge \bar{z}) = \bar{y} \wedge \bar{z} ,$$

$$(x \wedge y \wedge \bar{z}) \vee (x \wedge y \wedge z) = x \wedge y . \text{ Отсюда,}$$

$$(\bar{x} \wedge \bar{y} \wedge \bar{z}) \vee (x \wedge \bar{y} \wedge \bar{z}) \vee (x \wedge y \wedge \bar{z}) \vee (x \wedge y \wedge z) = (\bar{y} \wedge \bar{z}) \vee (x \wedge y) -$$

сокращенная ДНФ

Задания для практической работы: построить таблицу истинности, найти СНДФ, найти минимальную ДНФ для высказывания:

Вариант 1

1. $(\bar{z} \vee y) \rightarrow (\bar{z} \oplus \bar{x})$
2. $\left((\overline{A \wedge B}) \Rightarrow A \right) \Rightarrow A \vee B$
3. $(\bar{z} \vee y) \wedge (\bar{z} \oplus \bar{x})$

Вариант 2

1. $\left((\overline{A \wedge B}) \Rightarrow A \right) \Leftrightarrow (A \vee B)$
2. $x | (y \rightarrow z) \oplus (x | y) \rightarrow (x | z)$
3. $(\bar{z} \Rightarrow y) \Leftrightarrow (\bar{z} \vee \bar{x})$

Вариант 3

1. $(x | y) \rightarrow (x | z)$
2. $(\overline{A \wedge B}) \Leftrightarrow (\bar{B} \oplus \bar{A}) \Leftrightarrow (A \vee B) \oplus (A \oplus \bar{B})$
3. $(\bar{z} \oplus y) \Rightarrow (\bar{z} | (y \vee \bar{x}))$

Вариант 4

1. $(\overline{A \Rightarrow B}) \Leftrightarrow (\bar{B} \wedge \bar{A})$
2. $(x \wedge y) \oplus (x \wedge z) \Leftrightarrow x \wedge (y \oplus z)$
3. $(\bar{z} \oplus x) \vee (\bar{z} | (y \vee \bar{x}))$

Вариант 5

1. $((x \downarrow y) \rightarrow z) \oplus y$
2. $(x | y) \rightarrow (x | z) \oplus (\bar{z} \vee y) \rightarrow (\bar{z} \oplus \bar{x})$
3. $(\bar{z} \vee y) \rightarrow (\bar{z} | (y \vee \bar{x}))$

Вариант 6

1. $(x \vee \bar{y}) \rightarrow (\bar{z} \oplus \bar{x})$
2. $(\overline{A \Rightarrow B}) \Leftrightarrow (\bar{B} \wedge \bar{A}) \oplus ((A \Rightarrow B) \wedge \bar{B}) \Rightarrow A$
- 3.

Вариант 7

- 1.
2. $(\overline{A \Rightarrow B}) \vee (\bar{B} \wedge \bar{A}) \Rightarrow ((A \Rightarrow B) \wedge \bar{B}) \oplus A$
3. $(\bar{z} \vee x) \Leftrightarrow (\bar{z} | (y \vee \bar{x}))$

Вариант 8

1. $((A \vee B) \wedge B) \Rightarrow A$
2. $x[(y \Rightarrow z) \Leftrightarrow (x|y) \vee (x|z)]$
3. $(\bar{z} \Leftrightarrow y) \Leftrightarrow (\bar{z}|(y \oplus \bar{x}))$

Вариант 9

1. $(x|\bar{y}) \oplus (z \rightarrow \bar{x})$
2. $(\overline{A \Rightarrow B}) \vee (\bar{B} \wedge \bar{A}) \Leftrightarrow ((A \Rightarrow B) \oplus \bar{B}) \vee A$
3. $((A \vee B) \oplus \bar{B}) \Rightarrow A$

Вариант 10

1. $\left(A \vee B \wedge A \right) \Leftrightarrow A$
2. $(x \wedge y) \vee (x \wedge z) \Rightarrow$
3. $(x \vee \bar{y}) \rightarrow (\bar{z} \Leftrightarrow \bar{x})$

Вариант 11

1. $(\bar{z} \rightarrow x) \Leftrightarrow (y|x)$
2. $(\overline{A \vee B}) \vee (\bar{B} \wedge \bar{A}) \Leftrightarrow ((A \vee B) \oplus \bar{B}) \Rightarrow A$
3. $(\bar{z} \oplus y) \vee (\bar{z}|(y \vee \bar{x}))$

Вариант 12

- 1.
2. $x[(y \oplus z) \oplus (x|y) \vee (x|z)]$
3. $(\overline{A \vee B}) \Leftrightarrow (\bar{B} \wedge \bar{A})$

ПРАКТИЧЕСКАЯ РАБОТА №2

Тема: Закон алгебры логики.

Цель: изучить правила десятичной арифметики

Теоретический материал

Содержание работы

1. Организационный момент

- Проверка готовности учащихся к уроку.
- Приветствие.
- Проверка готовности ребят к уроку

2. Постановка темы и цели урока

3. Повторение изученного материала

Сложение	Вычитание	Умножение	Деление
$0 + 0 = 0$	$0 - 0 = 0$	$00 = 0$	$0:1 = 0$
$0 + 1 = 1$	$1 - 0 = 1$	$0 - 1 = 0$	$1:1 = 1$
$1 + 0 = 1$	$1 - 1 = 0$	$1 - 0 = 0$	
$1 + 1 = 0$ и «1» переноса в ст. разряд суммы	$10 - 1 = 1$ необходима «1» замена из ст. разряда	$1 - 1 = 1$	

Сложение. Рассмотрим сложение чисел в двоичной системе счисления. В его основе лежит таблица сложения одноразрядных двоичных чисел:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

Важно обратить внимание на то, что при сложении двух единиц происходит переполнение разряда и производится перенос в старший разряд. Переполнение разряда наступает тогда, когда величина числа в нем становится равной или большей основания.

Сложение многоразрядных двоичных чисел происходит в соответствии с вышеприведенной таблицей сложения с учетом возможных переносов из младших разрядов в старшие. В качестве примера сложим в столбик двоичные числа 110, и 11,:

$$\begin{array}{r} +110_2 \\ 11_2 \\ \hline 1001_2 \end{array}$$

Вычитание. Рассмотрим вычитание двоичных чисел. В его основе лежит таблица вычитания одноразрядных двоичных чисел. При вычитании из

меньшего числа (0) большего (1) производится заем из старшего разряда. В таблице заем обозначен 1 с чертой:

$$\begin{aligned} 0 - 0 &= 0, \\ 0 - 1 &= \overline{1}1, \\ 1 - 0 &= 1, \\ 1 - 1 &= 0. \end{aligned}$$

Вычитание многоразрядных двоичных чисел происходит в соответствии с вышеприведенной таблицей вычитания с учетом возможных заемов из старших

$$\begin{array}{r} -110_2 \\ 11_2 \\ \hline 11_2 \end{array}$$

разрядов.

В качестве примера произведем вычитание двоичных чисел 110, и 11₂:

Умножение. В основе умножения лежит таблица умножения одноразрядных двоичных чисел:

$$\begin{aligned} 0 \cdot 0 &= 0, \\ 0 \cdot 1 &= 0, \\ 1 \cdot 0 &= 0, \\ 1 \cdot 1 &= 1. \end{aligned}$$

Умножение многоразрядных двоичных чисел происходит в соответствии с вышеприведенной таблицей умножения по обычной схеме, применяемой в десятичной системе счисления, с последовательным умножением множимого на цифры множителя. В качестве примера произведем умножение двоичных чисел 110, и 11₂:

$$\begin{array}{r} \times 110_2 \\ 11_2 \\ \hline 110 \\ 110 \\ \hline 10010_2 \end{array}$$

Деление. Операция деления выполняется по алгоритму, подобному алгоритму выполнения операции деления в десятичной системе счисления. В качестве примера произведем деление двоичного числа 110, на 11₂:

$$\begin{array}{r|l} -110_2 & 11_2 \\ \hline 11 & 10_2 \\ \hline 0 & \end{array}$$

Арифметические операции над двоичными числами

В примерах для получения результата при умножении и сложении чисел применение таблиц «Сложение» и «Умножение» начинается с младших разрядов. При делении чисел использование таблицы «Вычитание» начинается со старших разрядов.

Таблица 4

Арифметические операции над двоичными числами

Сложение	Умножение	Деление	
0_2 $+1011$	X^{1011}_2	0011_2 -10	$11-$ 1
$юю_2$	$1010.$	111	101_2 1
$0,$ 10000	10111 ‘	0 -11	
	10111	111	
	10_2 111001	111	
		ш	
		0	

Любая математическая операция над двоичными числами требует выполнения команд «сложить», «вычесть», «сдвинуть» (вправо или влево).

Можно выполнять арифметические действия в восьмеричной и шестнадцатеричной системах счисления. Необходимо только помнить, что величина переноса в следующий разряд при сложении и заем из старшего разряда при вычитании определяются величиной основания системы

$$\begin{array}{r} +37_8 \\ 25_8 \\ \hline 64_8 \end{array} \quad \begin{array}{r} -9C_{16} \\ 78_{16} \\ \hline 24_{16} \end{array}$$

счисления:

Для проведения арифметических операций над числами, выраженными в различных системах счисления, необходимо предварительно перевести их в одну и ту же систему.

ЗАДАНИЕ 1

Контрольные вопросы по пройденному материалу

- 1. Что такое система счисления?
- 2. Как называется единица измерения количества информации?
- 3. Какая система счисления одной и той же цифре присваивает различное значение, определяющееся позицией в последовательности цифр, изображающей число?
- 4. Какая система счисления построена на позиционном принципе записи чисел с основанием 10?
- 5. Какая система счисления построена на позиционном принципе записи чисел с основанием 8?
- 6. Что такое триада? Где она используется?
- 7. Что такое тетрада? Где она используется?
- 8. В чем различие между триадой и тетрадой?
- 9. Сложить числа в шестнадцатеричной системе счисления: 2ADEC и FAEDA.
- 10. Умножить числа в восьмеричной системе счисления: 761 и 205.
- 11. Переведите число 703,04 из восьмеричной системы счисления в десятичную.
- 12. Переведите число B2E,4 из шестнадцатеричной системы счисления в десятичную.
- 13. Какие три команды выполняются при любых математических операциях над двоичными числами?

ПРАКТИЧЕСКАЯ РАБОТА №3

Тема: Основные логические элементы.

Цель работы: научиться переводить основные логические элементы.

Оборудование, приборы, аппаратура, материалы: персональный компьютер с выходом в Интернет.

Справочный материал 1,2

Содержание работы

1. Организационный момент
 - Проверка готовности учащихся к уроку.
 - Приветствие.
 - Проверка готовности ребят к уроку
2. Постановка темы и цели урока
3. Повторение изученного материала

1. Теоретические сведения

Логическое высказывание – это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

Математический аппарат алгебры логики очень удобен для описания того, как функционируют аппаратные средства компьютера, поскольку основной системой счисления в компьютере является двоичная, в которой используются цифры 1 и 0, а значений логических переменных тоже два: 1 и 0.

Логический элемент компьютера — это часть электронной логической схемы, которая реализует элементарную логическую функцию.

Логическими элементами компьютеров являются электронные схемы И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ и др. (называемые также вентилями), а также триггер. С помощью этих схем можно реализовать любую логическую функцию, описывающую работу устройств компьютера. Работу логических элементов описывают с помощью таблиц истинности.

Схемы, реализующие логические функции, называются логическими элементами. Основные логические элементы имеют, как правило, один выход (Y) и несколько входов, число которых равно числу аргументов ($X_1; X_2; X_3 \dots X_N$). На электрических схемах логические элементы обозначаются в виде прямоугольников с выводами для входных (слева) и выходных (справа) переменных. Внутри прямоугольника изображается символ, указывающий функциональное назначение элемента.

Схема И реализует конъюнкцию двух или более логических значений. Условное обозначение на структурных схемах схемы И с двумя входами представлено на рисунке а1.

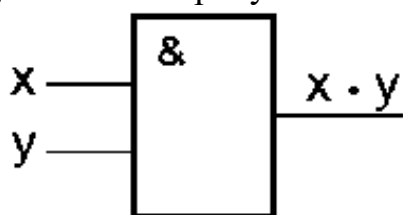


Рисунок 1 - Структурная схема схемы И

Таблица 1 - Таблица истинности схемы И

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Единица на выходе схемы И будет тогда и только тогда, когда на всех входах будут единицы. Когда хотя бы на одном входе будет ноль, на выходе также будет ноль.

Связь между выходом z этой схемы и входами x и y описывается соотношением: $z = x \cdot y$ (читается как "х и у"). Операция конъюнкции на структурных схемах обозначается знаком "&" (читается как "амперсэнд"), являющимся сокращенной записью английского слова and.

1.2 Схема или

Схема ИЛИ реализует дизъюнкцию двух или более логических значений. Когда хотя бы на одном входе схемы ИЛИ будет единица, на её выходе также будет единица.

Условное обозначение на структурных схемах схемы ИЛИ с двумя входами представлено на рисунке 2. Знак "1" на схеме — от устаревшего обозначения дизъюнкции как " ≥ 1 " (т.е. значение дизъюнкции равно единице, если сумма значений операндов больше или равна 1). Связь между выходом z этой схемы и входами x и y описывается соотношением: $z = x \vee y$ (читается как "х или у").

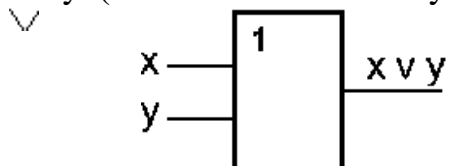


Рисунок 2 – Структурная схема схемы ИЛИ

Таблица 2 – Таблица истинности схемы ИЛИ

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

1.3 Схема не

Схема НЕ (инвертор) реализует операцию отрицания. Связь между

входом x этой схемы и выходом z можно записать соотношением $z=\bar{x}$, где \bar{x} читается как "не x " или "инверсия x ".

Если на входе схемы 0, то на выходе 1. Когда на входе 1, на выходе 0. Условное обозначение на структурных схемах инвертора представлена на рисунке 3.

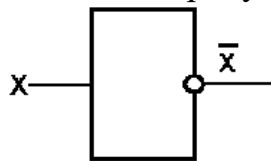


Рисунок 3 - Структурная схема инвертора

Таблица 3 - Таблица истинности схемы НЕ

x	\bar{x}
0	1
1	0

1.4 Схема и—не

Схема И—НЕ состоит из элемента И и инвертора и осуществляет отрицание результата схемы И. Связь между выходом z и входами x и y схемы записывают следующим образом: $z = \overline{x \cdot y}$, читается как "инверсия x и y ". Условное обозначение на структурных схемах схемы И—НЕ с двумя входами представлено на рисунке 4.

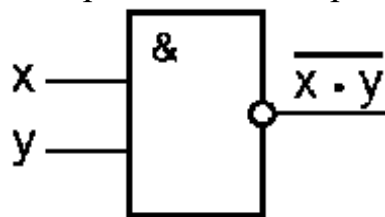


Рисунок 4 - Структурная схема схемы И—НЕ

Таблица 4 - Таблица истинности схемы И—НЕ

x	y	$\overline{x \cdot y}$
0	0	1
0	1	1
1	0	1
1	1	0

Схема ИЛИ—НЕ состоит из элемента ИЛИ и инвертора и осуществляет отрицание результата схемы ИЛИ. Связь между выходом z и входами x и y схемы записывают следующим образом: $z = \overline{x \vee y}$, читается как "инверсия x или y ". Условное обозначение на структурных схемах схемы ИЛИ—НЕ с двумя входами представлено на рисунок 5.

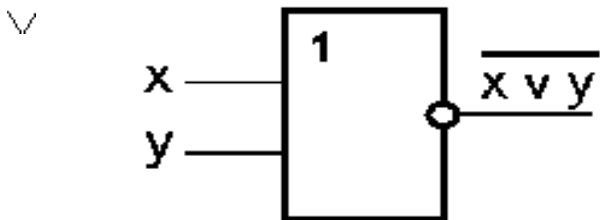


Рисунок 5 - Структурная схема схемы

ИЛИ—НЕ

Таблица 5 - Таблица истинности схемы ИЛИ—НЕ

x	y	$x \vee \overline{y}$
0	0	1
0	1	0
1	0	0
1	1	0

2. Пример выполнения заданий

Проверить тождественность логических функций X и Y

$$X = \neg(f_1 * f_2 + \neg f_1 * f_3) * (f_1 * \neg f_2 + f_2 * f_3) * (f_1 * f_3 + \neg f_2)$$

$$Y = (f_1 + f_2 * f_3) * (\neg f_1 + \neg f_2 * f_3) * (\neg f_1 * \neg f_2 + f_1 * f_3)$$

Таблица 6 – Значения функции x

1	2	3	$\neg(f_1 * f_2 + \neg f_1 * f_3)$	$(f_1 * \neg f_2 + f_2 * f_3)$	$(f_1 * f_3 + \neg f_2)$	x	
			$(0*0+1*0)$ 1	$(0*1+0*0)$) 0	$(0*0+1)$) 1	0	
			$(0*0+1*1)$ 0	$(0*1+0*1)$) 0	$(0*1+1)$) 1	0	
			$(0*1+1*0)$ 1	$(0*0+1*0)$) 0	$(0*0+0)$) 0	0	
			$(1*0+0*0)$ 1	$(1*1+0*0)$) 1	$(1*0+1)$) 1	1	
			$(0*1+1*1)$ 0	$(0*0+1*1)$) 1	$(0*1+0)$) 1	0	
			$(1*0+0*1)$ 1	$(1*1+0*1)$) 1	$(1*1+1)$) 1	1	
			$(1*1+0*0)$	$(1*0+1*0)$	$(1*0+0)$	0	

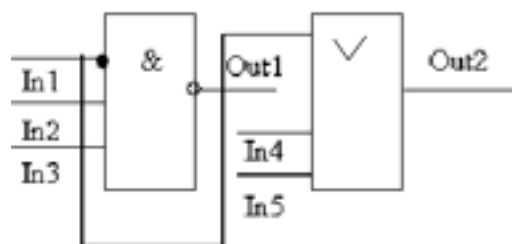
			0) 0) 0		
			$(1*1+0*1)$ 0	$(1*0+1*1)$) 1	$(1*1+0)$) 1	0	

Таблица 7 – Значения функции у

1	2	3	$(f_1+f_2*f_3)$	$(^f_1+^f_2*f_3)$	$(^f_1*^f_2+f_1*f_3)$	
			$(0+0*0)$ 0	$(1+1*0)$ 1	$(1*1+0*0)$ 1	
			$(0+0*1)$ 0	$(1+1*1)$ 1	$(1*1+0*1)$ 1	
			$(0+1*0)$ 0	$(1+0*0)$ 1	$(1*0+0*0)$ 0	
			$(1+0*0)$ 0	$(0+1*0)$ 0	$(0*1+1*0)$ 0	
			$(0+1*1)$ 1	$(1+0*1)$ 1	$(1*0+0*1)$ 0	
			$(1+0*1)$ 1	$(0+1*1)$ 1	$(0*1+1*1)$ 1	
			$(1+1*0)$ 0	$(0+0*0)$ 0	$(0*0+1*1)$ 0	
			$(1+1*1)$ 1	$(0+0*1)$ 0	$(0*0+1*2)$ 0	

Вывод: при одинаковых значениях на входе функций f_1, f_2, f_3 , значения функций x и y различны, соответственно x и y не тождественны.

Составить логическое выражение по схеме и таблицы истинности для выходных функций.



$$\text{Out 1} = \text{In1} * \text{In2} * \text{In3}$$

Таблица 8 – Значения функции Out1

ln1	ln2	ln3	$\overline{\ln 1}$	$\overline{\ln 1} * \ln 2 * \ln 3$	$\overline{\ln * \ln 2 * \ln 3}$
0	0	0	1	0	1
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	0	0	1

$$\text{Out2} = \ln 1 + \ln 4 + \ln 5$$

Таблица 9 – Значения функции Out2

ln1	ln4	ln5	$\overline{\ln 1}$	$\overline{\ln 1} + \ln 4 + \ln 5$
0	0	0	1	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	1

Проверить тождественность логических функций X и Y.

1.

$$X = (f1 + f2 * f3) * (f1 * f3 + \wedge f2) * (\wedge f1 * \wedge f2 + f1 * f3)$$

$$Y = (\wedge f1 + \wedge f2 * f3) * (f1 * f2 + \wedge f1 * f3) * (f1 * \wedge f2 + f2 * f3)$$

2.

$$X = (f1 + f2 * f3) * \wedge (\wedge f1 * \wedge f2 + f * f3) * (\wedge f1 + \wedge f2 * f3)$$

$$Y = \wedge (f1 * f3 + \wedge f2) * (f1 * \wedge f2 + f2 * f3) * (f1 * f2 + \wedge f1 * f3)$$

3.

$$X = (f1 + f2 * f3) * (\wedge f1 * \wedge f2 + f1 * f3) * \wedge (\wedge f1 + \wedge f2 * f3)$$

$$Y=(f_1^2+f_2^2f_3)^{f_1f_3+f_2}(f_1^2+f_2^2f_3)$$

4.

$$X=(f_1+f_2^2f_3)^{f_1^2+f_1f_3}(f_1^2+f_2^2f_3)$$

$$Y=(f_1^2+f_2^2f_3)^{f_1f_3+f_2}(f_1^2+f_2^2f_3) \quad 5.$$

$$X=(f_1^2+f_2^2f_3)^{f_1^2+f_1f_3}(f_1^2+f_2^2f_3)$$

$$Y=(f_1^2+f_2^2f_3)^{f_1f_3+f_2}(f_1^2+f_2^2f_3)$$

6.

$$X=(f_1+f_2^2f_3)^{f_1f_3+f_2}(f_1^2+f_2^2f_3)$$

$$Y=(f_1^2+f_2^2f_3)^{f_1^2+f_1f_3}(f_1^2+f_2^2f_3)$$

7.

$$X=(f_1+f_2^2f_3)^{f_1f_3+f_2}(f_1^2+f_2^2f_3)$$

$$Y=(f_1^2+f_2^2f_3)^{f_1^2+f_1f_3}(f_1^2+f_2^2f_3)$$

8.

$$X=(f_1^2+f_2^2f_3)^{f_1f_3+f_2}(f_1^2+f_2^2f_3)$$

$$Y=(f_1^2+f_2^2f_3)^{f_1^2+f_1f_3}(f_1^2+f_2^2f_3)$$

9.

$$X=(f_1^2+f_2^2f_3)^{f_1f_3+f_2}(f_1^2+f_2^2f_3)$$

$$Y=(f_1^2+f_2^2f_3)^{f_1^2+f_1f_3}(f_1^2+f_2^2f_3)$$

ПРАКТИЧЕСКАЯ РАБОТА №4

Тема: Триггеры.

Цель: изучить понятие триггер и способы его реализации

Оборудование, приборы, аппаратура, материалы: персональный компьютер с выходом в Интернет.

Справочный материал 1,2

Содержание работы

1. Организационный момент

- Проверка готовности учащихся к уроку.
- Приветствие.
- Проверка готовности ребят к уроку

2. Постановка темы и цели урока

3. Повторение изученного материала

ЗАДАНИЕ 1

Входы триггера

Входы триггеров обычно обозначают следующим образом:

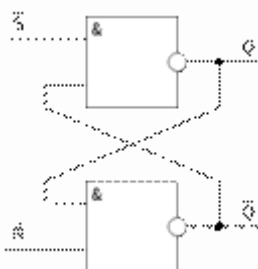
- S (от англ. Set, установить) — вход в RS-триггере;
- R (от англ. Reset, сброс) — вход в RS-триггере;
- J (от англ. Jump^[4], прыжок) — вход в JK-триггере;
- K (от англ. Kill, убить) — вход в JK-триггере;
- T (от англ. Toggles, переключить) — счётный вход в T-триггере;
- C (от англ. Clock, время) вход синхронизирующего сигнала. При

тактировании по фронту он часто обозначается стрелкой: стрелка внутрь — тактирование по переднему фронту, наружу — по заднему.

- D (от англ. Delay, задержка) — вход в D-триггере;
- E или EN (от англ. Enable, разрешить) — дополнительный асинхронный управляющий вход для разрешения приёма информации (иногда используют букву V).

Входы **J, K, T, D** всегда синхронные, т.е. тактируются по синхронизирующему сигналу на входе **C**. Разумеется, в каждом конкретном триггере имеются лишь некоторые из перечисленных входных линий. Входы **S** и **R** зачастую присутствуют не только в RS триггерах, но и в других типах триггеров, где предназначены, в основном, для асинхронного сброса устройства в 0 или установки в 1.

RS-триггер, или SR-триггер

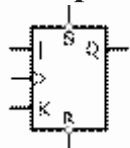


Одна из наглядных схем реализации асинхронного RS-триггера на базе двух элементов **2И-НЕ (NAND2)**

RS-триггер, или **SR-триггер** — триггер, который сохраняет своё предыдущее состояние при нулевых входах и меняет своё выходное состояние при подаче на один из его входов единицы. При подаче единицы на вход **S** (от англ. Set - установить) выходное состояние становится равным логической единице. А при подаче единицы на вход **R** (от англ. Reset - сбросить) выходное состояние становится равным логическому нулю. Если RS-триггер синхронный, то состояние его входов учитывается только в момент тактирования, например по переднему фронту импульса. Состояние, при котором на оба входа **R** и **S** одновременно поданы логические единицы, является запрещённым. Так, например, схема RS-триггера, изображённая на рисунке, при подаче на оба инверсных входа логического нуля перейдёт в состояние, когда на обоих выходах будут единицы, что не соответствует логике выхода триггера, поскольку инверсный выход \bar{Q} будет равен неинверсному Q , т.е. .

RS-триггер используется для создания сигнала с положительным и отрицательным фронтами, отдельно управляемыми посредством стробов, разнесённых во времени.

JK-триггер

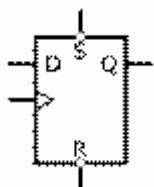


Символ JK-триггера с дополнительными асинхронными входами **S** и **R**, аналогично представлению в среде разработки

JK-триггер работает также как RS-триггер, с одним лишь исключением: при подаче логической единицы на оба входа **J** и **K** состояние выхода триггера изменяется на противоположное. Вход **J** (от англ. Jump - прыжок) аналогичен входу **S** у RS-триггера. Вход **K** (от англ. Kill - убить) аналогичен входу **R** у RS-триггера. При подаче единицы на вход **J** и нуля на вход **K** выходное состояние триггера становится равным логической единице. А при подаче единицы на вход **K** и нуля на вход **J** выходное состояние триггера становится равным логическому нулю. JK-триггер в отличие от RS-триггера не имеет запрещённых состояний на основных входах, однако это никак не помогает при нарушении правил разработки логических схем. На практике применяются только синхронные JK-триггеры, то есть состояния основных входов **J** и **K** учитываются только в момент тактирования, например по положительному фронту импульса на входе синхронизации.

На базе JK-триггера возможно построить D-триггер или T-триггер. Как можно видеть в таблице истинности JK-триггера, он переходит в инверсное состояние каждый раз при одновременной подаче на входы **J** и **K** логической 1. Это свойство позволяет создать на базе JK-триггера T-триггер, объединив входы **J** и **K**^[5].

D-триггер

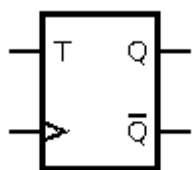


Символ D-триггера с дополнительными асинхронными входами **S** и **R**

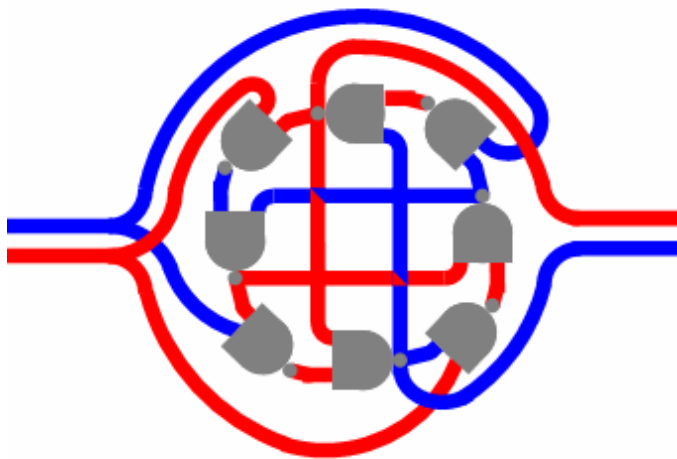
D-триггер (**D** от англ. delay - задержка) - запоминает состояние входа и выдаёт его на выход. D-триггеры имеют, как минимум, два входа: информационный **D** и синхронизации **C**. Сохранение информации в D-триггерах происходит в момент прихода активного фронта на вход **C**. Так как информация на выходе остаётся неизменной до прихода очередного импульса синхронизации, D-триггер называют также триггером с запоминанием информации или триггером-защёлкой. Рассуждая чисто теоретически, D-триггер можно образовать из любых RS- или JK-триггеров, если на их входы одновременно подавать взаимно инверсные сигналы.

D-триггер в основном используется для реализации защёлки. Так, например, для снятия 32 бит информации с параллельной шины, берут 32 D-триггера и объединяют их входы синхронизации для управления записью информации в защёлку, а 32 **D** входа подсоединяют к шине.

T-триггер



Изображение T-триггера на схемах.



Работа схемы T-триггера (при **T=1**) на базе восьми **2И-НЕ** логических вентилях. Слева — входы, справа — выходы. Синий цвет соответствует 0, красный — 1

T-триггер по каждому такту изменяет своё логическое состояние на противоположное при единице на входе **T**, и не изменяет выходное состояние при нуле на входе **T**. T-триггер часто называют счётным триггером. T-триггер может строиться как на JK, так и на D-триггерах. Как можно видеть в таблице

истинности JK-триггера, он переходит в инверсное состояние каждый раз при одновременной подаче на входы **J** и **K** логической 1. Это свойство позволяет создать на базе JK-триггера Т-триггер, объединяя входы **J** и **K**. Наличие в D-триггере динамического С входа позволяет получить на его основе Т-триггер. При этом вход D соединяется с инверсным выходом, а на вход С подаются счётные импульсы. В результате триггер при каждом счётном импульсе запоминает значение Q , то есть будет переключаться в противоположное состояние.

Т-триггер — триггер, который меняет своё предыдущее состояние на противоположное под воздействием каждого входного сигнала равного логической «1».

Характеристическое уравнение, определяющее логику функционирования асинхронного Т-триггера:

$$Q^{n+1} = T^n \overline{Q}^n \vee \overline{T}^n Q^n$$

Экспериментальное исследование работы асинхронного Т-триггера.

Для исследования работы Т-триггера используется модуль «Триггеры» лабораторного стенда. Если текущее состояние триггера Q_n не совпадает с требуемым состоянием согласно таблице истинности, то для переключения триггера необходимо подать единичный импульс или сигнал логической «1» на вход триггера.

Задание ответить на вопросы

КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Логический элемент —
- 2) Что такое Триггер?
- 3) Что такое Регистр?
- 4) Чем оперирует Триггер?
- 5) Чем оперирует Регистр?

ПРАКТИЧЕСКАЯ РАБОТА №5

Тема: Понятие архитектуры микропроцессора

Цель:

- изучить принцип организации современных микропроцессоров;
- изучить принцип функционирования современных микропроцессоров;
- изучить структуру современных микропроцессоров.

Результат обучения. После обучения студент должен:

- знать принцип организации и функционирования современных микропроцессоров;
- знать структуру современных микропроцессоров;
- уметь проводить анализ и классификацию различных процессоров.

План занятия:

1. Изучение теоретических вопросов темы 20 минут.
2. Выполнение практического задания 50 минут.
3. Контрольный опрос 10 минут.

Оборудование, приборы, аппаратура, материалы: персональный компьютер с выходом в Интернет.

Справочный материал 1,2

Содержание работы

1. Организационный момент
 - Проверка готовности учащихся к уроку.
 - Приветствие.
 - Проверка готовности ребят к уроку
2. Постановка темы и цели урока
3. Повторение изученного материала

1. Микропроцессор

Микропроцессор выполняют всю обработку информации в компьютере. Микропроцессор дешифрирует и выполняет команды программы, организует обращения к оперативной памяти, в нужных случаях инициирует работу периферийных устройств, воспринимает и обрабатывает запросы, поступающие из устройств машины и из внешней среды (“запросы прерывания”).

Для определений временных соотношений между различными этапами операции используется понятие машинного такта. Машинный такт определяет интервал времени, в течение которого выполняется одна или одновременно несколько микроопераций. Границы тактов задаются синхросигналами, вырабатываемыми специальной схемой — генератором синхросигналов. Также данная характеристика микропроцессора носит название тактовая частота, которая определяет, сколько микроопераций процессор выполнит за одну секунду.

2. Общая классификация микропроцессоров

В настоящее время насчитывается большое количество разнообразных процессоров. Приведем их общую классификацию.

По числу больших интегральных схем (БИС) в составе микропроцесса различают:

- однокристалльные микропроцессоры;
- многокристалльные микропроцессоры;
- многокристалльные секционные микропроцессоры.

Однокристалльные микропроцессоры получают при реализации всех аппаратных средств процессора в виде одной БИС или СБИС (сверхбольшой интегральной схемы). При усложнении микропроцессора приходится разбивать его на отдельные блоки. В этом случае каждый блок реализуется на отдельном кристалле, в результате чего процессор становится многокристалльным.

Многокристалльные секционные микропроцессоры получают в том случае, когда отдельные блоки процессора приходится логически разбивать дополнительно на секции. Секционность микропроцессоров дает возможность наращивать разрядность обрабатываемых данных или усложнять устройство управления микропроцессора.

По назначению различают:

- универсальные микропроцессоры;
- специализированные микропроцессоры.

Универсальные микропроцессоры могут быть применены для решения широкого круга разнообразных задач. Специализация МП, т.е. его проблемная ориентация на ускоренное выполнение определенных функций позволяет резко увеличить эффективную производительность при решении только определенных задач.

По виду обрабатываемых входных сигналов различают:

- цифровые микропроцессоры;
- аналоговые микропроцессоры.

Цифровые микропроцессоры работают с информацией представленной в виде числовых значений (дискретная форма). Аналоговые микропроцессоры работают с информацией, которая представлена в аналоговой форме, т.е. в виде непрерывного ряда значений.

По характеру временной организации работы микропроцессоры делятся на:

- синхронные микропроцессоры;
- асинхронные микропроцессоры.

Синхронные микропроцессоры – это микропроцессоры, в которых начало и конец выполнения операций задаются специальным устройством управления. Т.е. если в микропроцессоре присутствует устройство управления, то он относится к синхронным.

Асинхронные микропроцессоры позволяют начало выполнения каждой следующей операции определить по сигналу фактического окончания выполнения предыдущей операции.

По количеству выполняемых программ различают:

- однопрограммные микропроцессоры;
- многопрограммные микропроцессоры.

В однопрограммных микропроцессорах выполняется только одна программа. Переход к выполнению другой программы происходит после завершения текущей программы. В много- или мультипрограммных микропроцессорах одновременно выполняется несколько программ.

3. Архитектуры микропроцессоров

Существует две основные архитектуры современных процессоров – это архитектуры CISC и RISC. **CISC** (CISC - CompleteInstructionSetComputer) – это процессоры с полным набором команд, **RISC** (RISC - ReducedInstructionSetComputer) – это процессоры с сокращенным набором команд. Разберемся, чем одна архитектура отличается от другой.

Набор команд CISC был разработан для удобства программистов, которые вынуждены были писать программы для компьютеров на языке Ассемблер. Для ускорения процесса разработки программ в систему команд CISC были введены удобные команды, которые как бы представляли собой подпрограммы. В итоге, команды CISC-процессора имеют разную длину и время выполнения. К тому же CISC-процессор отличается невысокой производительностью, т.к. для выполнения некоторых команд требуется несколько машинных тактов.

В общем случае для **CISC-процессоров** характерно следующее:

- небольшое число регистров общего назначения;
- большое количество машинных команд, которые выполняются за много тактов;
- большое количество методов адресации;
- большое количество форматов команд различной разрядности;
- наличие команд обработки типа регистр-память.

К процессорам класса CISC относятся широко распространенные в персональных компьютерах процессоры фирм Intel, AMD, Cyrix.

В процессорах с набором команд **RISC** все команды имеют одинаковую длину и формат, а также простую адресацию памяти. Каждая команда выполняет только простые действия за один такт.

В общем случае для **RISC-процессоров** характерно следующее:

- отделение команд обработки данных от команд работы с памятью;
- выполнение любой команды занимает небольшое количество машинных тактов (предпочтительно один машинный такт);
- логика выполнения команд с целью повышения производительности ориентируется на аппаратную, а не на микропрограммную реализацию;
- используются команды фиксированной длины и фиксированного формата;
- наличие большого числа регистров, что позволяет большему объему данных храниться в регистрах на процессорном кристалле большее время. Это значительно увеличивает быстродействие процессора.

Проще говоря, сущность архитектуры RISC состоит в том, что в процессоре выполняются простые команды за один такт. При этом любую сложную команду можно разбить на несколько простых. Выполнение простых команд происходит быстрее, чем сложных, причем выполнение простых команд может происходить параллельно. Поэтому быстродействие RISC процессоров в общем случае выше, чем у CISC.

Считается, что в будущем процессоры с архитектурой RISC заменят менее перспективные процессоры с архитектурой CISC.

Существует еще одно понятие архитектуры процессоров, которые мы также рассмотрим. Наверняка вы часто встречались с термином «x86» (мы его несколько раз упомянули выше), или «Intel-совместимый процессор». Что за этим скрывается на самом деле? Современный x86-процессор – это процессор, способный исполнять машинный код архитектуры IA32 (архитектура 32-битных процессоров Intel). Этот код исполнял процессор Intel 80386 (известный как «386-й»). В настоящее время всё программное обеспечение для ПК разрабатывается именно для x86-процессоров. Оно выполняется на любом x86-процессоре, независимо от того, кто его произвел.

Кроме того, у архитектуры IA32 существуют дополнительные наборы команд от разработчика, компании Intel: MMX, SSE, SSE2 и SSE3. Также существуют неофициальные расширенные наборы команд: EMMX, 3DNow! и Extended 3DNow! – их разработала компания AMD.

Все перечисленные дополнительные наборы команд служат для увеличения быстродействия при выполнении некоторых операций. Одна команда из дополнительного набора, как правило, выполняет действие, для которого понадобилась бы небольшая программа, состоящая из команд основного набора.

4. Понятия irq и dma

Прерывание IRQ (InterruptReQuest - запрос прерывания) – это сигнал, по которому процессор узнает о совершении некоторого события, на которое необходимо “обратить” внимание. Пусть, к примеру, микропроцессор выполняет некоторую программу, и пусть в это время в каком-то внешнем устройстве произошло событие, на которое нужно обратить внимание, (например, на клавиатуре нажата клавиша). Естественно, ждать пока закончится выполнение текущей программы нельзя, она может работать еще долго и за это время может быть нажато много других клавиш, так что информация о первой из нажатых клавиш будет потеряна. Надо сразу, оперативно прореагировать на это событие.

Получив сигнал прерывания, микропроцессор прерывает выполнение текущей последовательности команд, а вместо нее начинает выполнять другую последовательность, соответствующую данному прерыванию.

Все прерывания делятся на три группы:

- аппаратные прерывания;
- логические прерывания;
- программные прерывания.

Аппаратные прерывания связаны с запросами от внутренних или периферийных устройств. Логические возникают при работе самого микропроцессора. Программные инициируются выполняемой программой.

Для IBM PC AT на базе процессоров Pentium предусмотрено было **16 линий IRQ**, часть которых заняты внутренними устройствами, а остальные используется внешними или не используются. В настоящее время число прерываний составляет несколько десятков.

Таким образом, число периферийных устройств, подключаемых к персональному компьютеру с использованием прерываний IRQ, не может превышать пяти.

DMA (DirectMemoryAccess) – это режим прямого доступа к памяти, когда периферийное устройство связано с оперативной памятью компьютера непосредственно, минуя микропроцессор. Этот режим наиболее эффективен, когда требуется высокая скорость обмена при передаче большого количества информации.

На IBM PC AT есть **8** независимых каналов DMA. Каналы DMA распределены следующим образом:

- 0 - микропроцессор;
- 1 - не используется;
- 2 - контроллер флоппи-диска;
- 3 - не используется;
- 4 - не используется;
- 5 - не используется;
- 6 - не используется;
- 7 - не используется.

Таким образом, к ПК можно подключить 5 различных устройств, которые используют режим DMA. При этом следует помнить, что не все устройства, требующие применения прерываний IRQ, используют DMA.

Практическое задание

1. Используя свой персональный компьютер (или его макет) определите модель используемого микропроцессора в вашем персональном компьютере. Запишите **ответ в отчет**.
2. Определите фирму-производителя микропроцессора. Запишите **ответ в отчет**.
3. Определите тактовую частоту микропроцессора. Запишите ответ в отчет.
4. Определите установочный разъем микропроцессора (можете использовать Интернет для поиска информации). Запишите **ответ в отчет**.
5. Самостоятельно проведите классификацию имеющегося микропроцессора. Запишите **ответ в отчет**.
6. Загрузите ПК. Вызовите программу **Сведения о системе** (Пуск – Программы – Стандартные – Служебные или файл **MSINFO32.EXE**).
7. Используя программу **Сведения о системе**, выпишите в **отчет** общее число прерываний IRQ вашего компьютера.

8. **Выпишите в отчет** основные устройства, которые используют прерывания IRQ вашего компьютера.
9. **Укажите в отчете**, сколько свободных прерываний есть в вашем компьютере.
10. Выпишите в отчет все занятые каналы DMA вашего компьютера.
11. Выпишите в отчет все свободные каналы DMA вашего компьютера.
12. Определите, какой процессор в настоящее время является наиболее оптимальным при выборе компьютера для дома, для выполнения графических работ, для офисной работы. Обоснуйте и докажите свой ответ.

ПРАКТИЧЕСКАЯ РАБОТА №6

Тема: Понятие архитектуры микропроцессора

Цель:

- изучить принцип организации современных микропроцессоров;
- изучить принцип функционирования современных микропроцессоров;
- изучить структуру современных микропроцессоров.

Результат обучения. После обучения студент должен:

- знать принцип организации и функционирования современных микропроцессоров;
- знать структуру современных микропроцессоров;
- уметь проводить анализ и классификацию различных процессоров.

План занятия:

4. Изучение теоретических вопросов темы 20 минут.
5. Выполнение практического задания 50 минут.
6. Контрольный опрос 10 минут.

Оборудование, приборы, аппаратура, материалы: персональный компьютер с выходом в Интернет.

Справочный материал 1,2

Содержание работы

1. Организационный момент
 - Проверка готовности учащихся к уроку.
 - Приветствие.
 - Проверка готовности ребят к уроку
2. Постановка темы и цели урока
3. Повторение изученного материала

Практическое задание

1. Загрузите ПК. Вызовите программу **Сведения о системе** (**Пуск – Программы – Стандартные – Служебные** или файл **MSINFO32.EXE**).
2. Используя программу **Сведения о системе**, выпишите в отчет общее число прерываний **IRQ** вашего компьютера.
3. **Выпишите в отчет** основные устройства, которые используют прерывания **IRQ** вашего компьютера.
4. **Укажите в отчете**, сколько свободных прерываний есть в вашем компьютере.
5. Выпишите в отчет все занятые каналы **DMA** вашего компьютера.
6. Выпишите в отчет все свободные каналы **DMA** вашего компьютера.
7. Определите, какой процессор в настоящее время является наиболее оптимальным при выборе компьютера для дома, для выполнения графических работ, для офисной работы. Обоснуйте и докажите свой ответ.

ПРАКТИЧЕСКАЯ РАБОТА №7

Тема: Комбинационные схемы.

Цель работы: изучение методов синтеза комбинационных схем в различных универсальных базисах логических элементов.

Оборудование, приборы, аппаратура, материалы: персональный компьютер с выходом в Интернет.

Справочный материал 1,2

Содержание работы

1. Организационный момент

- Проверка готовности учащихся к уроку.
- Приветствие.
- Проверка готовности ребят к уроку

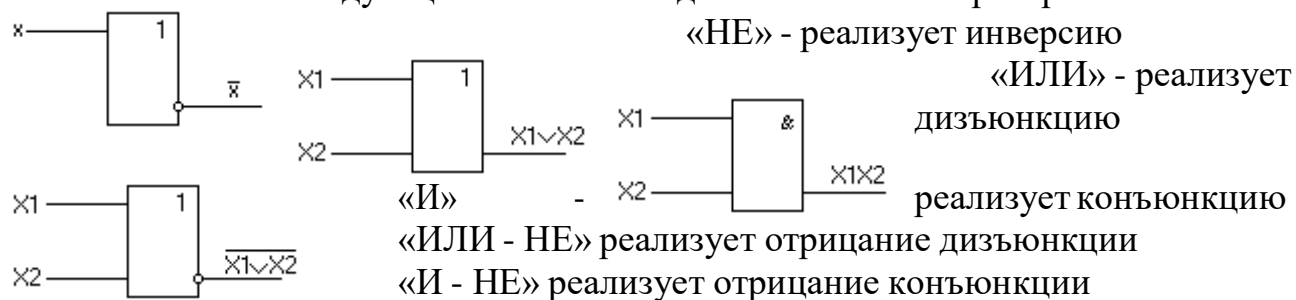
2. Постановка темы и цели урока

3. Повторение изученного материала

1. Теоретическая часть

1.1. Понятие о комбинационной схеме и логическом операторе

Комбинационной схемой (КС) называется схема, характеризующаяся наличием сигналов на входах и выходах в каждый момент времени. Комбинационных схем в составе ЭВМ множество: это любая схема, на входы которой подаются сигналы «1» и «0» и такие же сигналы снимаются с выходов. Для описания КС используются **логические операторы**, т.е. элементарные логические функции (ЛФ), реализуемые схемой. Для описания КС используют следующие основные виды логических операторов:



Использование логических операторов позволяет решать задачи анализа и синтеза комбинационных схем.

Задача анализа КС – это задача получения логической функции, описывающей работу схемы. Эта функция в дальнейшем может использоваться для упрощения КС.

1.2. Синтез комбинационных схем

1.2.1. Канонический метод

Задача синтеза КС – это построение реальной схемы по физическому описанию, под которым понимается описание принципа работы схемы. Для синтеза КС чаще всего используется канонический метод.

Алгоритм метода:

5. К объединенным клеткам применяется правило склеивания: при объединении 2-х клеток освобождается одна координата, 4-х – две, 8-ми – три и т.д.

$x_3 x_4 x_5$		$x_1 x_2$							
		000	001	011	010	110	111	101	100
$x_1 x_2$	00	0	1	3	2	6	7	5	6
	01	8	9	11	10	14	15	13	12
	10	24	25	27	26	30	31	29	28
	11	16	17	19	18	22	23	21	20

x_5

x_4

x_3

x_5

x_2

x_1

Рис.1.1. Карта Карно для функции пяти переменных

В случае шести переменных потребуется уже четыре карты Карно для четырех переменных. Эта карта состоит из 64 клеток (рис.1.2).

Метод минимизирующих карт, также как и метод Квайна-Мак-Класки, относится к точным методам. Классический метод Квайна – Мак-Класки при большом числе переменных требует значительных временных затрат, так как требуется определить все первичные импликанты функции. Существует группа методов минимизации, позволяющих находить точное решение задачи, избегая получения всех первичных импликант.

$x_4 x_5 x_6$		$x_1 x_2 x_3$							
		000	001	011	010	110	111	101	100
$x_1 x_2 x_3$	000	0	1	3	2	6	7	5	4
	001	8	9	11	10	14	15	13	12
	011	24	25	27	26	30	31	29	28
	010	16	17	19	18	22	23	21	20
	110	48	49	51	50	54	55	53	52
	111	56	57	59	58	62	63	61	60
	101	40	41	43	42	46	47	45	44
	100	32	33	35	34	38	39	37	36

x_6

x_5

x_4

x_6

x_3

x_2

x_1

Рис.1.2. Карта Карно для функции шести переменных

1.2.3. Метод сжатия булевой матрицы

Логическая функция, заданная в СДНФ, представляется в матричной форме.

Метод сжатия булевой матрицы заключается в замене некоторой совокупности строк этой матрицы, образующей интервал булева пространства, соответствующим троичным вектором.

Если задана булева матрица **B**, то в ней находят определяющие элементы. Для этого берется первый элемент матрицы **B**, отыскиваются все соседние с ним элементы и строится содержащий эти элементы интервал (троичный вектор), называемый минимальным поглощающим интервалом. Если все элементы этого интервала содержатся в **B**, то интервал будет максимальным и обязательным.

Минимальный поглощающий интервал для совокупности элементов булева пространства определяется так: строится троичный вектор, компоненты которого принимают значения одноименных компонент булевых векторов, если эти значения совпадают, и значение «-», если среди значений одноименных компонент булевых векторов встречаются как 0, так и 1. Для определения того, является ли найденный интервал максимальным, нужно путем перебора установить, все ли его элементы являются элементами матрицы **B**.

Пример. Рассмотрим булеву матрицу **B**:

$$B = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Будем анализировать подряд элементы этой матрицы, начав с элемента с номером 1:

0 0 0 1 0 1 1.

Этот элемент имеет трех соседей:

0 0 0 0 0 1 2,

0 1 0 1 0 1 10,

1 0 0 1 0 1 20,

минимальный поглощающий интервал для которых (- - 0 - 0 1) не содержится в **B** (например, там отсутствует элемент (1 1 0 1 0 1), принадлежащий данному интервалу). Следовательно, элемент 1 не является определяющим.

Такой же результат получается и для следующих элементов, вплоть до элемента 5:

0 1 1 0 1 0 5,

который оказывается определяющим. Действительно, соседними с ним являются элементы

0 1 1 1 1 0 4,

0 1 0 0 1 0 9,

1 1 1 0 1 0 17,

и минимальный поглощающий интервал для них $(-1 - -1 0)$ полностью содержится в матрице **B**. В этом можно убедиться, найдя в **B** элемент

$$\begin{matrix} 1 & 1 & 0 & 1 & 1 & 0 & 14, \end{matrix}$$

ортогональный элементу 5 по компонентам 1, 3 и 4, вместе с его соседями по этим компонентам

$$\begin{matrix} 0 & 1 & 0 & 1 & 1 & 0 & 8, \end{matrix}$$

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 & 0 & 16, \end{matrix}$$

$$\begin{matrix} 1 & 1 & 0 & 0 & 1 & 0 & 15. \end{matrix}$$

Исключив из текущей матрицы **B*** (равной вначале **B**) перечисленные восемь элементов интервала $(-1 - -1 0)$, принимаемого в качестве первого элемента кратчайшего покрытия **B**, продолжим перебор в матрице **B***.

Следующим определяющим оказывается элемент

$$\begin{matrix} 1 & 1 & 0 & 1 & 0 & 0 & 13, \end{matrix}$$

который вместе с соседями

$$\begin{matrix} 1 & 1 & 0 & 0 & 0 & 0 & 12, \end{matrix}$$

$$\begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 & 18, \end{matrix}$$

$$\begin{matrix} 1 & 0 & 0 & 1 & 0 & 0 & 19. \end{matrix}$$

образует поглощающий интервал $(1 - 0 - 0 0)$, который полностью содержится в **B**.

Интервал $(1 - 0 - 0 0)$ оказывается максимальным и принимается в качестве второго элемента решения, а текущая матрица **B*** сокращается еще на четыре элемента и приводится к следующему виду:

$$B^* = \begin{matrix} \begin{matrix} 00 & 0 & 0 & 1 & 0 & 10 & 1 \\ 00 & 0 & 0 & 0 & 0 & 10 & 2 \\ 00 & 1 & 1 & 1 & 0 & 00 & 3 \\ 00 & 1 & 1 & 1 & 0 & 10 & 6 \\ 00 & 1 & 0 & 0 & 0 & 00 & 7 \\ 00 & 1 & 0 & 1 & 0 & 10 & 10 \\ 00 & 1 & 0 & 0 & 0 & 00 & 11 \\ 01 & 0 & 0 & 1 & 0 & 10 & 20 \\ 01 & 0 & 0 & 0 & 0 & 10 & 21 \end{matrix} \end{matrix}$$

Следующий в порядке перебора элемент 20 также оказывается определяющим, и в решение включается соответствующий ему допустимый интервал $(-0 0 - 0 1)$, образуемый элементами 1, 2, 20 и 21, исключаемыми в связи с этим из текущей матрицы **B***.

Затем перебор идет на второй круг, и последовательно находятся определяющие элементы 3,7,10, соответствующие которым допустимые интервалы $(0 1 1 1 0 -)$, $(0 1 0 0 0)$, $(0 - 0 - 0 1)$ также включаются в решение. Текущая матрица **B*** сокращается при этом до пустого множества, а совокупность найденных последовательно допустимых интервалов образует решение рассматриваемой задачи, представляемое троичной матрицей **T**:

$$T = \begin{matrix} \begin{matrix} 0- & 1 & - & - & 1 & 0 \\ 01 & - & 0 & - & 0 & 0 \\ 0- & 0 & 0 & - & 0 & 1 \\ 00 & 1 & 1 & 1 & 0 & - \\ 00 & 1 & 1 & & 0 & - \\ 00 & - & 0 & - & 0 & 1 \end{matrix} \end{matrix}$$

1.2.4. Синтез КС в универсальных базисах И-НЕ, ИЛИ-НЕ, И-ИЛИ-НЕ

Разрабатываемые комбинационные схемы(КС), используемые в устройствах ЭВМ, могут иметь разнообразную конфигурацию. Задача реализации КС состоит в преобразовании описывающих её логических функций в суперпозицию логических элементов заданного типа. При этом предполагается, что исходная функция должна быть представлена в минимальной форме: МДНФ или МКНФ, а на входах проектируемой схемы вместе с переменными присутствует их инверсия.

Правила преобразования исходной функции для рациональной реализации на элементах И-НЕ, ИЛИ-НЕ, И-ИЛИ-НЕ:

1. Для реализации исходной функции на элементах типа И-НЕ необходимо от МДНФ функции взять двойное отрицание и одно из них раскрыть по правилу де Моргана.

2. Для реализации исходной булевой функции на элементах типа ИЛИ-НЕ необходим от МКНФ функции взять двойное отрицание и одно из них раскрыть по правилу де Моргана

3. Для реализации исходной булевой функции на элементах типа И-ИЛИ-НЕ необходимо найти МДНФ отрицания функции.

Пример 1. Реализовать функцию $f(x_1, x_2, x_3, x_4)$ (0, 2, 4, 6, 7, 10, 11, 14, 15) на элементах: \vee =

а) типа И-НЕ;

б) типа ИЛИ-НЕ;

в) типа И-ИЛИ-НЕ.

Решение: а) Для реализации на элементах типа И-НЕ с помощью карты Карно (рис.1.3, а) получим МДНФ функции :

$$f = x_1 x_2 \vee x_2 x_3 \vee \bar{x}_1 \bar{x}_4.$$

Берём двойное отрицание от МДНФ функции и одно из них открываем по правилу де Моргана :

$$f = \overline{\overline{x_1 x_2 \vee x_2 x_3 \vee \bar{x}_1 \bar{x}_4}} = \overline{\overline{x_1 x_2} \vee \overline{x_2 x_3} \vee \overline{\bar{x}_1 \bar{x}_4}}.$$

б) Реализация функции на элементах типа И-НЕ показана на рис. 1.4 а, б. Для реализации функции f на элементах типа ИЛИ-НЕ получаем МКНФ функции с помощью карт Карно (рис. 1.3, б):

$$f = (\bar{x}_1 \vee x_3) \cdot (x_3 \vee \bar{x}_4) \cdot (x_1 \vee x_2 \vee \bar{x}_4).$$

Берём двойное отрицание от МКНФ и одно из них открываем по правилу де Моргана:

$$f = \overline{\overline{(\bar{x}_1 \vee x_3) \cdot (x_3 \vee \bar{x}_4) \cdot (x_1 \vee x_2 \vee \bar{x}_4)}} = \overline{\bar{x}_1 \vee x_3 \vee x_3 \vee \bar{x}_4 \vee x_1 \vee x_2 \vee \bar{x}_4}.$$

Реализация функции на элементах типа ИЛИ-НЕ показана на рис. 1.4,б.

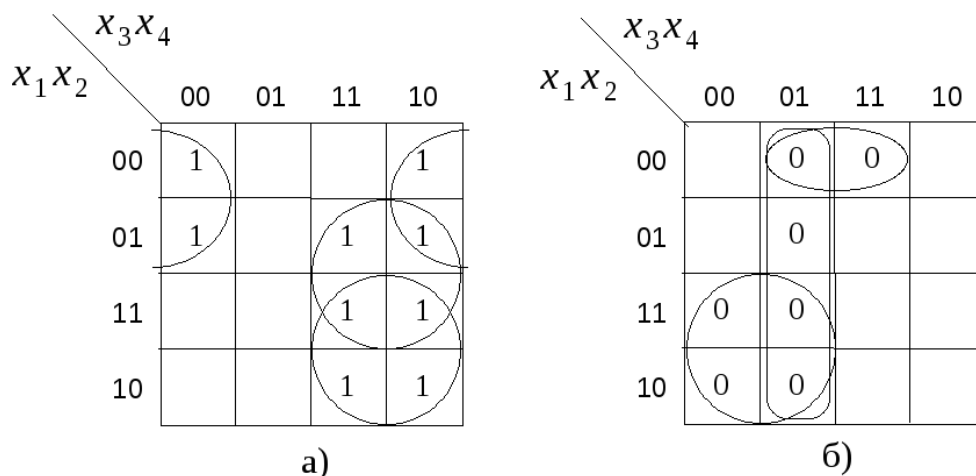


Рис. 1.3. Карта Карно функции к примеру 1

в) Для реализации функции f на элементах типа И-ИЛИ-НЕ находим с помощью карт Карно (см. рис. 1.3, б) МДНФ отрицания функции:

$$f = \overline{x_1 \bar{x}_3} \vee \overline{\bar{x}_3 x_4} \vee \overline{\bar{x}_1 \bar{x}_2 x_4}.$$

Реализация полученной функции приведена на рис. 1.4, в.

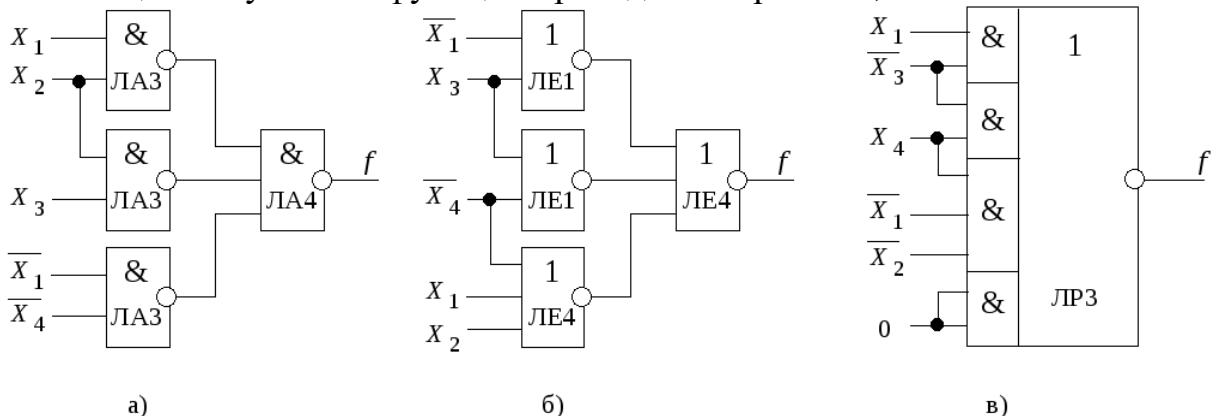


Рис.1.4. Реализация схемы из примера 1: а - на элементах типа И-НЕ; б - на элементах типа ИЛИ-НЕ; в - на элементах типа И-ИЛИ-НЕ

Некоторые приемы преобразования функций для реализации на элементах заданного типа

1) При реализации довольно сложных булевых функций указанными приемами иногда получаются выражения, которые непосредственно не реализуются на элементах заданного типа. В этом случае применяют метод тождественных преобразований с предварительной группировкой.

Пример 2. Реализовать функцию Y (x_1, x_2, x_3, x_4, x_5) на элементах типа И-ИЛИ-НЕ серии К555:

$$Y = x_1 \bar{x}_2 x_4 \vee x_1 \bar{x}_3 \bar{x}_5 \vee x_2 \bar{x}_4 x_5 \vee \bar{x}_3 \bar{x}_4 x_5.$$

Решение. С помощью карты Карно (рис.1.5,а) находим МДНФ отрицания функции Y , склеивая нули:

$$Y = \overline{\bar{x}_1 x_4} \vee \overline{x_1 \bar{x}_5} \vee \overline{x_2 x_4 x_5} \vee \overline{x_2 x_3 \bar{x}_5} \vee \overline{\bar{x}_2 x_3 \bar{x}_4}.$$

Полученное выражение не реализуется непосредственно на ЛЭ типа И-ИЛИ-НЕ, имеющих в серии К555. Сгруппируем два последних терма и получим:

$$Y = \overline{\bar{x}_1 x_4} \vee \overline{x_1 \bar{x}_5} \vee \overline{x_2 x_4 x_5} \vee \overline{x_3 \cdot (x_2 \bar{x}_5 \vee \bar{x}_2 \bar{x}_4)} = \overline{\bar{x}_1 x_4} \vee \overline{x_1 \bar{x}_5} \vee \overline{x_2 x_4 x_5} \vee \overline{x_3 A},$$

где $A = x_2 \bar{x}_5 \vee \bar{x}_2 \bar{x}_4$.

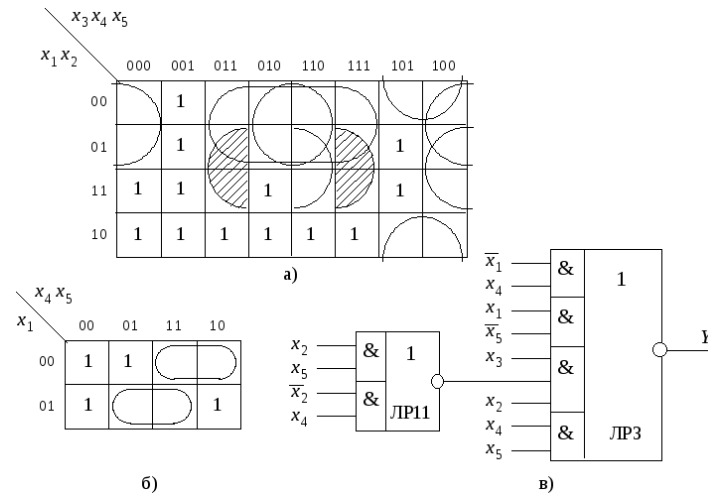


Рис. 1.5. Карты Карно (а, б) и схема (в) к примеру 2.

Для $A (x_2 x_4 x_5)$ еще раз применим карту Карно (рис. 1.5, б), с целью приведения этой функции к форме отрицания ДНФ, получим: $A = \overline{x_2 x_5} \vee \overline{\bar{x}_2 x_4}$.

В итоге получили функцию, легко реализуемую на элементах типа И-ИЛИ-НЕ: К555 ЛР11 и К555 ЛР3 (рис. 1.5, в).

2) При синтезе КС на основе реальных ЛЭ необходимо учитывать ограниченное число их входов и нагрузочную способность. Для синтеза КС на ЛЭ с меньшим числом входов, чем это требуется по исходной булевой функции, используется факторизационный метод синтеза КС или метод каскадов.

Этот метод базируется на преобразовании исходной булевой функции путем выноса за скобку общих логических переменных. В этом случае схема реализации булевой функции будет состоять из более, чем двух уровней, что приведет к уменьшению ее быстродействия.

Пример 3. Реализовать функцию $Y (x_1, x_2, x_3)$ на элементах типа 2И-НЕ (К555 ЛА3).

$$Y = \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3.$$

Решение. Преобразуем функцию Y в суперпозицию заданных ЛЭ следующим образом:

$$Y = \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3 = \bar{x}_1 (\bar{x}_2 x_3 \vee x_2 \bar{x}_3) \vee x_1 (\bar{x}_2 \bar{x}_3 \vee x_2 x_3) \\ = \bar{x}_1 \cdot A \vee x_1 \cdot \bar{A},$$

$$\text{где } A = \bar{x}_2 x_3 \vee x_2 \bar{x}_3.$$

От функции Y и A берем двойное отрицание, одно из которых раскрываем по правилу де Моргана:

$$Y = \overline{\overline{\bar{x}_1 A} \vee \overline{x_1 \bar{A}}} = \overline{\overline{\bar{x}_1 A} \cdot \overline{x_1 \bar{A}}}; \quad A = \overline{\overline{\bar{x}_2 x_3} \vee \overline{x_2 \bar{x}_3}} = \overline{\overline{\bar{x}_2 x_3} \cdot \overline{x_2 \bar{x}_3}}.$$

Полученные выражения позволяют реализовать схему с использованием элементов ЛА3 серии К555 (рис. 1.6.).

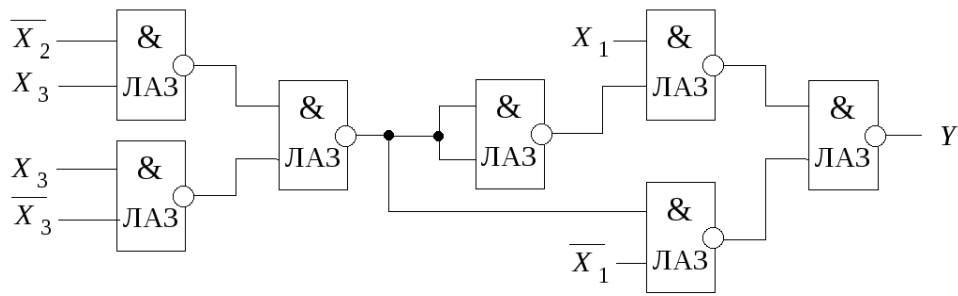


Рис. 1.6. Схема к примеру 3

3) В ряде случаев требуется построить КС, не используя инверсии входных переменных. С этой целью выполняются преобразования исходной функции, рассмотренные в примере 4.

Пример 4. Реализовать функцию $Y(x_1, x_2)$ на элементах серии К555, не используя инверсии входных переменных:

$$Y = x_1 \bar{x}_2 \vee \bar{x}_1 x_2.$$

Решение. Используя правила поглощения для элементарных дизъюнкций, можно записать:

$$Y = x_1 \bar{x}_2 \vee \bar{x}_1 x_2 = x_1 (\bar{x}_1 \vee \bar{x}_2) \vee x_2 (\bar{x}_1 \vee \bar{x}_2).$$

Далее берем от функции Y двойное отрицание, одно из которых раскрываем по правилу де Моргана, получим:

$$Y = x_1 (\bar{x}_1 \vee \bar{x}_2) \vee x_2 (\bar{x}_1 \vee \bar{x}_2) = \overline{x_1 x_1 x_2} \cdot \overline{x_2 x_1 x_2}.$$

Реализация этой функции на элементах ЛА3 серии К555 показана на рис. 1.7.

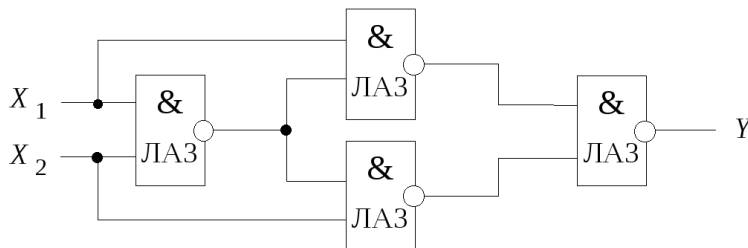


Рис. 1.7. Схема к примеру 4

8, рекомендуется способ преобразования функции, рассмотренный в примере 5. ≥4) В том случае, если ДНФ функции содержит число элементарных конъюнкций

Пример 5. Реализовать функцию

$$Y(x_1, x_2, x_3, x_4, x_5, x_6) = \bar{x}_1 \bar{x}_3 \vee \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_6 \vee \bar{x}_1 x_2 x_6 \vee \bar{x}_3 \bar{x}_6 \vee \bar{x}_3 \bar{x}_4 x_6 \vee x_5 x_6 \vee x_2 x_4 \bar{x}_6 \vee \bar{x}_1 \bar{x}_4 \bar{x}_5 \bar{x}_6 \vee x_1 x_2 x_3 x_5.$$

Решение. Сгруппируем исходные термы так, чтобы число слагаемых в скобке было равно (или <) числу схем И в ЛЭ типа И-ИЛИ-НЕ, а ранг соответствовал числу входов схем И:

$$Y = (\bar{x}_1 \bar{x}_3 \vee \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_6 \vee \bar{x}_1 x_2 x_6) \vee (\bar{x}_3 \bar{x}_6 \vee \bar{x}_3 \bar{x}_4 x_6 \vee x_5 x_6 \vee x_2 x_4 \bar{x}_6) \vee (\bar{x}_1 \bar{x}_4 \bar{x}_5 \bar{x}_6 \vee x_1 x_2 x_3 x_5).$$

Возьмем от функции Y двойное отрицание, одно из которых раскроем, не меняя выражения в скобках, получим:

$$Y = \overline{\bar{x}_1 \bar{x}_3 \vee \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_6 \vee \bar{x}_1 x_2 x_6} \cdot \overline{\bar{x}_3 \bar{x}_6 \vee \bar{x}_3 \bar{x}_4 x_6 \vee x_5 x_6 \vee x_2 x_4 \bar{x}_6}.$$

$$\overline{x_1} \overline{x_4} \overline{x_5} \overline{x_6} \vee x_1 x_2 x_3 x_5 .$$

Реализация этой функции показана на рис. 1.8.

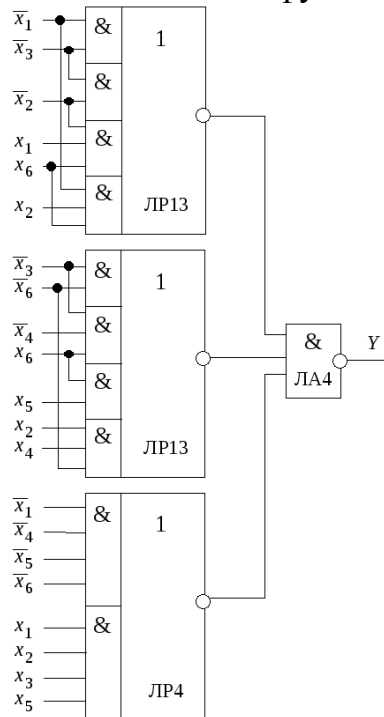


Рис. 1.8. Схема к примеру 5

1.2.5. Методы синтеза КС с несколькими выходами

Это задача построения КС, реализующей систему ЛФ(1.1).

Для синтеза используются методы:

1. **декомпозиции,**
2. **классический,**
3. **каскадов.**

Метод декомпозиции заключается в построении комбинационных схем, реализующих каждое из уравнений системы.

Комбинационная схема, соответствующая (1.1), будет состоять из k отдельных КС. Недостаток метода – избыточная сложность схемы.

Классический метод имеет следующий алгоритм:

1. Любым классическим методом определяются первичные импликанты функций y_1, y_2, \dots, y_k .
2. Функции y_1, y_2, \dots, y_k выражаются через найденные импликанты.
3. Строится КС, соответствующая функциям, найденным в пункте 2.

Метод каскадов (факторизации) заключается в преобразовании функции n переменных: $f = x_n f(x_1, x_2, \dots, x_{n-1}) \vee \overline{x_n} f(x_1, x_2, \dots, x_{n-1})$, которое выполняется над всеми ЛФ системы (1.1). Разложение по переменным выполняется до тех пор, пока не будет получена функция минимального числа переменных. (см. пример 3, п.1.2.4.).

1.2.6. Оценка качества комбинационных схем

Качество КС оценивается по двум параметрам: **сложности и быстродействию.**

Сложность КС – это количество элементов, используемых для её построения. Однако такая оценка эффективна только для реальных схем, при проектировании чаще используют оценку сложности **по Квайну**. В соответствии с этой оценкой сложность КС определяется как суммарное число входов всех элементов схемы. Для оценки по Квайну можно использовать ДНФ исходной функции. Каждая переменная в ДНФ оценивается как «1», инверсная переменная – как «2»; к этому числу прибавляется количество дизъюнкций в выражении плюс 1 на каждое дизъюнктивное выражение. При оценке сложности в корпусах за единицу измерения берётся 1/12 или 1/14 корпуса. Подсчитывается общее число выводов всех элементов, т.е. всех входов и всех выходов, и делится на 12 или 14.

Быстродействие КС определяется как максимальное время прохождения сигнала от входа схемы к её выходу. При оценке быстродействия считают, что все элементы имеют приблизительно одинаковую задержку τ . Быстродействие оценивается как $T = \tau * N$, где N – количество элементов на максимальном пути от входа схемы к её выходу. Следует отметить, что скобочная форма ЛФ уменьшает количество элементов схемы, но увеличивает время прохождения сигнала (уменьшает быстродействие).

Задания к выполнению работы

Задание выполняется в соответствии с выбранным вариантом (см. табл.) в следующем порядке:

Логическую функцию, заданную в числовом виде, представить в СДНФ и матричной форме.

Выполнить минимизацию методом минимизирующих карт, методом Квайна - Мак-Класки и методом сжатия булевой матрицы.

5, составить программу минимизации или использовать известные средства автоматизации, ознакомившись с литературой. ≥ Поскольку исходные ЛФ имеют число переменных

Минимизированную ЛФ представить в заданном универсальном базисе И-НЕ или ИЛИ-НЕ.

ЗАДАНИЕ 1

Ответить на контрольные вопросы

Таблица

		0,1,2,3,6,8,10,12,17,19,21,23,27,29	И- НЕ
		0,1,4,5,8,9,10,11,17,20,21,22,23,30,31	И- НЕ
		4,5,6,7,11,12,15,17,21,23,24,26,28	И- НЕ
		4,6,8,10,11,13,14,15,16,17,25,26,27,28	И- НЕ

		2,3,4,5,10,11,16,17,20,22,24,26,29,31	И- НЕ
		4,5,10,12,14,16,21,23,27,29,46,47,50,51	И- НЕ
		6,7,12,13,14,15,21,25,27,31,37,39,55,57	И- НЕ
		4,8,12,13,20,24,27,28,36,38,42,44,60,62	И- НЕ
		1,3,5,7,12,13,14,15,30,31,36,38,40,48,52	И- НЕ
0		6,8,12,13,20,21,22,23,27,29,35,37,47,49	И- НЕ
1		1,5,6,7,8,9,10,11,16,18,20,22,24,26,28,29	И- НЕ
2		1,2,5,6,8,9,12,13,14,15,21,23,25,27,29,31	И- НЕ
3		3,7,8,9,10,11,14,15,17,18,19,21,26,28	И- НЕ
4		3,4,5,7,8,9,10,14,15,16,17,20,22,24,26	И- НЕ
5		2,3,5,6,10,11,12,13,16,18,21,22,25,26,28	И- НЕ
6	9	6,8,12,13,14,15,17,18,19,23,29,31,45,47,4	ИЛ И-НЕ
7	1,43	12,13,16,17,19,20,21,22,24,27,29,35,37,4	ИЛ И-НЕ
8	46	0,4,5,7,14,15,22,24,30,32,34,36,38,42,44,	ИЛ И-НЕ
9	39	5,7,8,9,12,13,18,24,25,27,29,31,33,35,37,	ИЛ И-НЕ
0	8	10,11,16,18,20,21,26,28,30,32,35,36,37,3	ИЛ И-НЕ
		6,7,10,11,14,15,17,19,21,23,26,27,28,29,3	ИЛ

1		1	И-НЕ
2		6,7,8,9,12,13,16,18,19,23,24,25,27,29,31	ИЛ И-НЕ
3		9 8,9,10,11,14,15,18,20,21,22,23,24,26,27,2	ИЛ И-НЕ
4		1 8,9,12,13,16,17,21,23,24,26,27,28,29,30,3	ИЛ И-НЕ
5		,29 7,8,9,10,11,12,14,17,19,20,21,22,23,25,27	ИЛ И-НЕ
6		,40 0,4,6,10,12,20,22,24,26,28,30,32,34,36,38	ИЛ И-НЕ
7		44 0,2,5,7,15,17,19,21,30,31,33,35,36,37,38,	ИЛ И-НЕ
8		43 0,3,4,7,20,21,22,23,32,33,34,35,40,41,42,	ИЛ И-НЕ
9		51,52 0,1,6,8,12,14,16,18,26,28,30,31,32,33,50,	ИЛ И-НЕ
0		63 0,1,2,3,10,11,16,17,24,28,48,49,50,51,62,	ИЛ И-НЕ
1		1,5,7,9,20,21,22,23,25,27,29,30,31	И- НЕ
2		0,1,2,3,10,11,12,16,17,18,20,27,28	И- НЕ

ПРАКТИЧЕСКАЯ РАБОТА №8

Тема: Конечные автоматы.

Цель:

- изучить принцип организации современных микропроцессоров;
- изучить принцип функционирования современных микропроцессоров;
- изучить структуру современных микропроцессоров.

Результат обучения. После обучения студент должен:

- знать принцип организации и функционирования современных микропроцессоров;
- знать структуру современных микропроцессоров;
- уметь проводить анализ и классификацию различных процессоров.

Оборудование, приборы, аппаратура, материалы: персональный компьютер с выходом в Интернет.

Справочный материал 1,2

Содержание работы

1. Организационный момент
 - Проверка готовности учащихся к уроку.
 - Приветствие.
 - Проверка готовности ребят к уроку
2. Постановка темы и цели урока
3. Повторение изученного материала

1. Микропроцессор

Микропроцессор выполняют всю обработку информации в компьютере. Микропроцессор дешифрирует и выполняет команды программы, организует обращения к оперативной памяти, в нужных случаях инициирует работу периферийных устройств, воспринимает и обрабатывает запросы, поступающие из устройств машины и из внешней среды (“запросы прерывания”).

Для определений временных соотношений между различными этапами операции используется понятие машинного такта. Машинный такт определяет интервал времени, в течение которого выполняется одна или одновременно несколько микроопераций. Границы тактов задаются синхросигналами, вырабатываемыми специальной схемой — генератором синхросигналов. Также данная характеристика микропроцессора носит название тактовая частота, которая определяет, сколько микроопераций процессор выполнит за одну секунду.

2. Общая классификация микропроцессоров

В настоящее время насчитывается большое количество разнообразных процессоров. Приведем их общую классификацию.

По числу больших интегральных схем (БИС) в составе микропроцессора различают:

- однокристалльные микропроцессоры;
- многокристалльные микропроцессоры;
- многокристалльные секционные микропроцессоры.

Однокристалльные микропроцессоры получают при реализации всех аппаратных средств процессора в виде одной БИС или СБИС (сверхбольшой интегральной схемы). При усложнении микропроцессора приходится разбивать его на отдельные блоки. В этом случае каждый блок реализуется на отдельном кристалле, в результате чего процессор становится многокристалльным.

Многокристалльные секционные микропроцессоры получают в том случае, когда отдельные блоки процессора приходится логически разбивать дополнительно на секции. Секционность микропроцессоров дает возможность наращивать разрядность обрабатываемых данных или усложнять устройство управления микропроцессора.

По назначению различают:

- универсальные микропроцессоры;
- специализированные микропроцессоры.

Универсальные микропроцессоры могут быть применены для решения широкого круга разнообразных задач. Специализация МП, т.е. его проблемная ориентация на ускоренное выполнение определенных функций позволяет резко увеличить эффективную производительность при решении только определенных задач.

По виду обрабатываемых входных сигналов различают:

- цифровые микропроцессоры;
- аналоговые микропроцессоры.

Цифровые микропроцессоры работают с информацией представленной в виде числовых значений (дискретная форма). Аналоговые микропроцессоры работают с информацией, которая представлена в аналоговой форме, т.е. в виде непрерывного ряда значений.

По характеру временной организации работы микропроцессоры делятся на:

- синхронные микропроцессоры;
- асинхронные микропроцессоры.

Синхронные микропроцессоры – это микропроцессоры, в которых начало и конец выполнения операций задаются специальным устройством управления.

Т.е. если в микропроцессоре присутствует устройство управление, то он относится к синхронным.

Асинхронные микропроцессоры позволяют начало выполнения каждой следующей операции определить по сигналу фактического окончания выполнения предыдущей операции.

По количеству выполняемых программ различают:

- однопрограммные микропроцессоры;
- многопрограммные микропроцессоры.

В однопрограммных микропроцессорах выполняется только одна программа. Переход к выполнению другой программы происходит после завершения текущей программы. В много- или мультипрограммных микропроцессорах одновременно выполняется несколько программ.

3. Архитектуры микропроцессоров

Существует две основные архитектуры современных процессоров – это архитектуры CISC и RISC. **CISC** (CISC - CompleteInstructionSetComputer) – это процессоры с полным набором команд, **RISC** (RISC - ReducedInstructionSetComputer) – это процессоры с сокращенным набором команд. Разберемся, чем одна архитектура отличается от другой.

Набор команд CISC был разработан для удобства программистов, которые вынуждены были писать программы для компьютеров на языке Ассемблер. Для ускорения процесса разработки программ в систему команд CISC были введены удобные команды, которые как бы представляли собой подпрограммы. В итоге, команды CISC-процессора имеют разную длину и время выполнения. К тому же CISC-процессор отличается невысокой производительностью, т.к. для выполнения некоторых команд требуется несколько машинных тактов.

В общем случае для **CISC-процессоров** характерно следующее:

- небольшое число регистров общего назначения;
- большое количество машинных команд, которые выполняются за много тактов;
- большое количество методов адресации;
- большое количество форматов команд различной разрядности;
- наличие команд обработки типа регистр-память.

К процессорам класса CISC относятся широко распространенные в персональных компьютерах процессоры фирм Intel, AMD, Cyrix.

В процессорах с набором команд **RISC** все команды имеют одинаковую длину и формат, а также простую адресацию памяти. Каждая команда выполняет только простые действия за один такт.

В общем случае для **RISC-процессоров** характерно следующее:

- отделение команд обработки данных от команд работы с памятью;

- выполнение любой команды занимает небольшое количество машинных тактов (предпочтительно один машинный такт);
- логика выполнения команд с целью повышения производительности ориентируется на аппаратную, а не на микропрограммную реализацию;
- используются команды фиксированной длины и фиксированного формата;
- наличие большого числа регистров, что позволяет большему объему данных храниться в регистрах на процессорном кристалле большее время. Это значительно увеличивает быстродействие процессора.

Проще говоря, сущность архитектуры RISC состоит в том, что в процессоре выполняются простые команды за один такт. При этом любую сложную команду можно разбить на несколько простых. Выполнение простых команд происходит быстрее, чем сложных, причем выполнение простых команд может происходить параллельно. Поэтому быстродействие RISC процессоров в общем случае выше, чем у CISC.

Считается, что в будущем процессоры с архитектурой RISC заменят менее перспективные процессоры с архитектурой CISC.

Существует еще одно понятие архитектуры процессоров, которые мы также рассмотрим. Наверняка вы часто встречались с термином «**x86**» (мы его несколько раз упомянули выше), или «Intel-совместимый процессор». Что за этим скрывается на самом деле? Современный x86-процессор – это процессор, способный исполнять машинный код архитектуры IA32 (архитектура 32-битных процессоров Intel). Этот код исполнял процессор Intel 80386 (известный как «386-й»). В настоящее время всё программное обеспечение для ПК разрабатывается именно для x86-процессоров. Оно выполняется на любом x86-процессоре, независимо от того, кто его произвел.

Кроме того, у архитектуры IA32 существуют дополнительные наборы команд от разработчика, компании Intel: MMX, SSE, SSE2 и SSE3. Также существуют неофициальные расширенные наборы команд: EMMX, 3DNow! и Extended 3DNow! – их разработала компания AMD.

Все перечисленные дополнительные наборы команд служат для увеличения быстродействия при выполнении некоторых операций. Одна команда из дополнительного набора, как правило, выполняет действие, для которого понадобилась бы небольшая программа, состоящая из команд основного набора.

4. Понятия irq и dma

Прерывание IRQ (InterruptReQuest - запрос прерывания) – это сигнал, по которому процессор узнает о совершении некоторого события, на которое необходимо “обратить” внимание. Пусть, к примеру, микропроцессор выполняет некоторую программу, и пусть в это время в каком-то внешнем устройстве произошло событие, на которое нужно обратить внимание,

(например, на клавиатуре нажата клавиша). Естественно, ждать пока закончится выполнение текущей программы нельзя, она может работать еще долго и за это время может быть нажато много других клавиш, так что информация о первой из нажатых клавиш будет потеряна. Надо сразу, оперативно прореагировать на это событие.

Получив сигнал прерывания, микропроцессор прерывает выполнение текущей последовательности команд, а вместо нее начинает выполнять другую последовательность, соответствующую данному прерыванию.

Все прерывания делятся на три группы:

- аппаратные прерывания;
- логические прерывания;
- программные прерывания.

Аппаратные прерывания связаны с запросами от внутренних или периферийных устройств. Логические возникают при работе самого микропроцессора. Программные инициируются выполняемой программой.

Для IBM PC AT на базе процессоров Pentium предусмотрено было **16 линий IRQ**, часть которых заняты внутренними устройствами, а остальные используются внешними или не используются. В настоящее время число прерываний составляет несколько десятков.

Таким образом, число периферийных устройств, подключаемых к персональному компьютеру с использованием прерываний IRQ, не может превышать пяти.

DMA (DirectMemoryAccess) – это режим прямого доступа к памяти, когда периферийное устройство связано с оперативной памятью компьютера непосредственно, минуя микропроцессор. Этот режим наиболее эффективен, когда требуется высокая скорость обмена при передаче большого количества информации.

На IBM PC AT есть **8** независимых каналов DMA. Каналы DMA распределены следующим образом:

- 0 - микропроцессор;
- 1 - не используется;
- 2 - контроллер флоппи-диска;
- 3 - не используется;
- 4 - не используется;
- 5 - не используется;
- 6 - не используется;
- 7 - не используется.

Таким образом, к ПК можно подключить 5 различных устройств, которые используют режим DMA. При этом следует помнить, что не все устройства, требующие применения прерываний IRQ, используют DMA.

Практическое задание

1. Используя свой персональный компьютер (или его макет) определите модель используемого микропроцессора в вашем персональном компьютере. Запишите **ответ в отчет**.
2. Определите фирму-производителя микропроцессора. Запишите **ответ в отчет**.
3. Определите тактовую частоту микропроцессора. Запишите ответ в отчет.
4. Определите установочный разъем микропроцессора (можете использовать Интернет для поиска информации). Запишите **ответ в отчет**.
5. Самостоятельно проведите классификацию имеющегося микропроцессора. Запишите **ответ в отчет**.
6. Загрузите ПК. Вызовите программу **Сведения о системе** (Пуск – Программы – Стандартные – Служебные или файл **MSINFO32.EXE**).
7. Используя программу **Сведения о системе**, выпишите в **отчет** общее число прерываний IRQ вашего компьютера.
8. **Выпишите в отчет** основные устройства, которые используют прерывания IRQ вашего компьютера.
9. **Укажите в отчете**, сколько свободных прерываний есть в вашем компьютере.
10. Выпишите в отчет все занятые каналы DMA вашего компьютера.
11. Выпишите в отчет все свободные каналы DMA вашего компьютера.
12. Определите, какой процессор в настоящее время является наиболее оптимальным при выборе компьютера для дома, для выполнения графических работ, для офисной работы. Обоснуйте и докажите свой ответ.

ПРАКТИЧЕСКАЯ РАБОТА №9

Тема: Работа арифметико-логических устройств

Цель работы: разобраться в арифметико логических устройствах.

Оборудование, приборы, аппаратура, материалы: персональный компьютер с выходом в Интернет.

Справочный материал 1,2

Подготовка к выполнению практической работы

1. Изучить описание практической работы.
2. Ответить на все контрольные вопросы.
3. Произвести синтез параллельного АЛУ по варианту (таблица 3).
4. Разработать функциональную схему параллельного АЛУ.
5. Разработать алгоритмы выполнения операций.
6. Выполнить операции над двоичными данными (кодами), согласно списка указанного в таблице вариантов (таблица 3).

Порядок выполнения работы

1. Произвести исследование i -го разряда комбинационной части АЛУ.
2. Произвести исследование АЛУ.

Отчет должен содержать

1. Функциональную схему АЛУ со списком операций и таблицей значений управляющих сигналов.
2. Таблицу установки признаков.

Контрольные вопросы

1. Назовите этапы синтеза АЛУ?
2. Назначение АЛУ.
3. Для чего служат флаговые регистры?
4. Как производится запись признаков во флаговый регистр?
5. Назначение буферных регистров A и A' , B и B' в схеме АЛУ?
6. Для чего используется схема MS при синтезе АЛУ?
7. Как производится использование сумматора при выполнении арифметической операции сложения?
8. Как используется сумматор при выполнении арифметической операции вычитания?
9. Как используется сумматор при выполнении арифметической операции декремент?
10. Как используется сумматор при выполнении арифметической операции инкремент?

ЗАДАНИЕ 1

Таблица 3.

N/N n/n	Число Разрядов АЛУ	Список команд	Коды операндов	
			А	В
0	3	$A+B, A-1, A-B, A \oplus$	0	011
1	4	B, \overline{A}	11	0111
2	2	$A-1, A \vee B, A/B,$	0	11
3	4	$A+B+1, B$	110	0011
4	2	$A-B, B-1, B+1, \overline{A} \cdot B,$	0	01
5	3	A	1	101
6	4	$A-B+1, A \oplus B, A \downarrow B$	0	0010
7	3	$, \overline{A}$	111	011
8	4	$B-1, A+B, A \cdot \overline{B}, A$	1	0101
9	2	$B+1, A-B, A \oplus B, A$	0	10
10	3	$A+B, A-B, B-1, A/B,$	0	010
11	2	B	11	01
12	4	$A-1, A+B+1, A \vee B, A$	1	0111
13	2	$A-B, A+B+1,$	110	10
14	3	$B+1, \overline{A} \cdot B, B$	0	011
15	4	$B-1, A+B+1, A \cdot \overline{B}, B$	11	0101
16	3	$A-1, A+B-1, A \cdot B, \overline{A},$	1	110
17	4	\overline{B}	111	0101
18	2	$B-1, A+B+1, A/B, A$	1	01
19	4	$B+1, A-B+1, A \oplus$	1	0011
20	3	B, \overline{A}	1	011
		$A-B, A+B+1,$	11	
		$B+1, A \downarrow B, B$	0	
		$B+1, A-B, A+B, B$	1	
		$A+B, A-B, B+1, A \cdot B,$	0	
		\overline{A}	101	
		$A+1, A-B-1, A \oplus B,$	1	
		$\overline{A} \cdot B, A$	0	
		$B-1, A+B+1, A/B, B$	11	
		$A-B, A+B+1, A \vee B, A-$	1	
		B, B	010	
			0	
			11	
			0	
			011	
			1	

			<div>1</div> <div>0</div> <div>111</div> <div>1</div> <div>01</div>	
--	--	--	---	--

ПРАКТИЧЕСКАЯ РАБОТА №10

Тема: Алгоритмы управления

Цель: закрепление навыков программного управления исполнителем алгоритмов.

Оборудование, программы: ПК, исполнитель «Стрелочка».

Содержание работы

1. Организационный момент

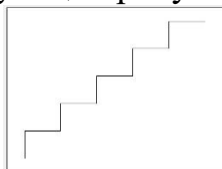
- Проверка готовности учащихся к уроку.
- Приветствие.
- Проверка готовности ребят к уроку

2. Постановка темы и цели урока

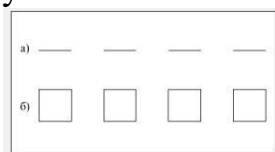
3. Повторение изученного материала

ЗАДАНИЕ 1

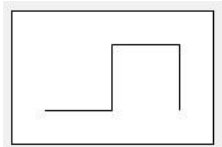
1. С помощью исполнителя алгоритмов стрелочка выполните следующий рисунок.



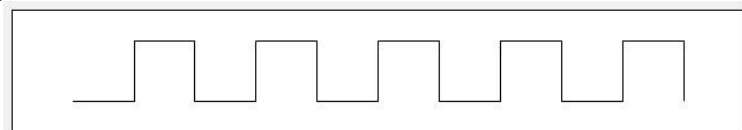
2. Написать программы для рисования следующих рисунков на всю длину листа с использованием вспомогательных алгоритмов (подпрограмм).



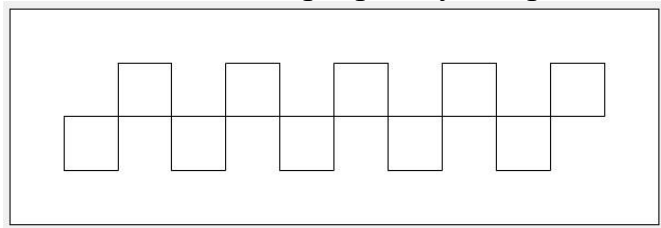
3. Написать подпрограмму для изображения следующей фигуры



4. Используя подпрограмму из предыдущего задания, отобразить следующий элемент.



5. Составить программу для рисования следующей фигуры



ПРАКТИЧЕСКАЯ РАБОТА №11

Тема: Процедура выполнения команд.

Цель: Изучить структуру и группы команд, выполняемые процессором УМК. Инструкция по выполнению работы

Оборудование, программы: ПК, исполнитель «Стрелочка».

Содержание работы

1. Организационный момент
 - Проверка готовности учащихся к уроку.
 - Приветствие.
 - Проверка готовности ребят к уроку
2. Постановка темы и цели урока
3. Повторение изученного материала

Теоретическая часть

Программа помещается в оперативную память вместе с данными. Каждая команда хранится в ячейке или группе ячеек и имеет свой адрес.

Команда – это элементарная инструкция машине, выполняемая ею автоматически без каких либо дополнительных указаний и пояснений.

Все команды имеют одинаковую структуру. Машинная команда состоит из двух частей: операционной и адресной

Код ОПерации (КОП)	Адреса
--------------------	--------

1. Код операции (операционная часть) определяет, какую команду нужно выполнить

2. Операнд (адресная часть) – над чем выполняется операция

Операционная часть команды (КОП) – это группа разрядов в команде, предназначенная для представления кода операции машине.

Адресная часть – это группа разрядов в команде, в которых записывают коды адресов операндов.

Операнд может быть простым и составным. Он может содержать в себе адресную часть, которая определяет, где хранятся данные и куда поместить результат операции.

В зависимости от количества используемых адресов, различают одноадресные, двух-, трех-, четырехадресные и безадресные команды.

- Одноадресные команды. Здесь указывается, где находится одно из чисел, второе должно быть помещено в АЛУ. Для этого существуют специальные команды пересылки данных.

- Двухадресные команды. Оба операнда в памяти, их адреса указаны. Результат заносится по одному из адресов.

- Трехадресные команды. Два адреса являются операндами, а третий служит для помещения туда результата.

- Четырехадресные команды. Два адреса – операнды, третий – результат, четвертый – адрес следующей команды.

- Безадресные команды. используются для выполнения служебных операций (очистить экран, заблокировать клавиатуру и так далее).

Команды выполняются друг за другом, а также могут быть выполнены ветвления и циклы.

Группы команд микропроцессора

Семейство микропроцессоров фирмы Intel от 8086 до Pentium имеет базовую систему команд, в состав которых входят:

Команды пересылки данных:

внутри процессора (Mov, Push, Pop и др.)

ввода-вывода (In, Out)

Арифметические команды:

основные (сложение, вычитание, умножение, деление)

дополнительные (INC, DEC)

Логические команды:

сдвиг, дизъюнкция, конъюнкция, и/или и др.

Обработка строковых данных:

пересылка, сканирование, сравнение, слияние.

Передачи управления:

безусловный переход, условный переход, прерывания, переход с возвратом.

Управления:

нет операции, внешняя синхронизация и так далее.

Каждая команда имеет много модификаций, чаще всего определяемых режимом адресации операндов.

Типы операндов команд

- Регистровые операнды указываются именами используемых регистров процессора.

- Непосредственные являются всегда числовыми величинами и могут быть в различных системах счисления. Числа различаются по последней букве, сопровождающей число:

b – двоичное число;

q – восьмеричное число;

d – десятичное число;

h – шестнадцатеричное число.

- Операнды в памяти могут указываться с помощью адресов ячеек, символическими именами, константами.

Различные комбинации этих элементов в команде называются способами адресации. Например, команда mov (переслать числа) может иметь следующие способы адресации:

ov	m	r,
ov	m	r,
ov	m	m,
ov	m	im
ov	m	ed, r
ov	m	im
ov	m	ed, m
ov	m	Sr,
ov	m	m
ov	m	Sr,
ov	m	r
ov	m	m,
ov	m	Sr
ov	m	r,
ov	m	Sr

Где r – РОН (регистр общего назначения); m – адрес в памяти;

imed – число; Sr – c

Задание 1

1. Изучить материал теоретической части;
2. Составить программу для условия: переписать 10000 байтов начиная с адреса А в другое место памяти начиная с адреса В. Оба эти имени относятся к сегменту данных, на начало которого указывает регистр As;
3. Записать микропрограмму суммирования двух чисел;
4. Сделать вывод о проделанной работе.

Информационное обеспечение обучения

Информационное обеспечение обучения

Печатные и электронные издания

Основные учебные издания:

1. Алфёров, В. В. Вычислительная техника и сети в отрасли: учебное пособие

/ В. В. Алфёров, Ю. М. Миронов. — Москва: Московская государственная академия водного транспорта, 2018. — 152 с. — ISBN 2227-8397. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/67596>

Дополнительные учебные издания:

2. Тюрин, И. В. Вычислительная техника: учебное пособие / И. В. Тюрин. — Тамбов: Тамбовский государственный технический университет, ЭБС АСВ, 2019.

— 112 с. — ISBN 978-5-8265-2099-4. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/99754>

Интернет ресурсы

3. <http://www.inf1.info/book/export/html/210> - Логические основы ЭВМ

4. <http://cssblok.ru/computer/prinsip.html> - Основные принципы функционирования компьютеров

5. <http://infolike.narod.ru/logic.html> - Основы алгоритмизации

Электронно-библиотечная система:

6. ЭБС «elibrary», ООО «РУНЭБ»
7. ЭБС «IPRbooks», ООО «Ай Пи Ар Медиа»
8. ЭБС «Лань», ООО «Издательство Лань»
9. ЭБС «PROФобразование»
10. ЭБС «Book.ru»